

【特許請求の範囲】**【請求項 1】**

デジタル署名のシェアを生成するコンピュータ実装方法であって、参加者のグループの各参加者は、第 1 の共有秘密鍵のそれぞれのシェアを有し、前記方法は、前記グループの第 1 の参加者によって実行され、

第 1 のメッセージを取得することと、

少なくとも第 1 の外部データ項目のハッシュに基づいて第 1 のデータ項目を生成することと、

エフェメラル秘密鍵の第 1 のエフェメラル秘密鍵シェアを生成することであって、前記第 1 のエフェメラル秘密鍵シェアは、前記第 1 のデータ項目と、各他の参加者によって生成されたそれぞれのデータ項目とに基づいて生成される、ことと、

前記エフェメラル秘密鍵に対応するエフェメラル公開鍵を生成することと、

前記第 1 のメッセージと、前記第 1 のエフェメラル秘密鍵シェアと、前記第 1 の共有秘密鍵の第 1 のシェアと、前記エフェメラル公開鍵とに基づいて、第 1 の署名シェアを生成することと、

少なくともしきい値数のそれぞれの署名シェアに基づいて第 1 の署名を生成するために、前記第 1 の署名シェアをコーディネータが利用できるようにすることと

を含む方法。

【請求項 2】

前記第 1 のエフェメラル秘密鍵シェアを生成することは、前記他の参加者の各々との秘密分散方式を実行することを含む、請求項 1 に記載の方法。

【請求項 3】

前記秘密分散方式は、共同検証可能秘密分散（J V R S S）方式である、請求項 2 に記載の方法。

【請求項 4】

前記第 1 のデータ項目は、第 1 の多項式のゼロ次係数であり、前記 J V R S S 方式は、前記第 1 の多項式のそれぞれのインスタンスを前記他の参加者の各々に送信することであって、前記第 1 の多項式の前記それぞれのインスタンスは、前記それぞれの参加者のそれぞれのインデックスに基づいて生成される、ことと、

各他の参加者からそれぞれの多項式を取得することであって、前記それぞれの多項式は、前記第 1 の参加者のそれぞれのインデックスと、前記他の参加者によって生成された前記それぞれのデータ項目とに基づいて生成される、ことと

を含む、請求項 3 に記載の方法。

【請求項 5】

前記エフェメラル秘密鍵に対応する前記エフェメラル公開鍵を生成することは、

前記第 1 のデータ項目に対応する第 1 の公開鍵を生成することと、

各他の参加者から、その他の参加者によって生成された前記それぞれのデータ項目に対応するそれぞれの公開鍵を取得することと

を含む、請求項 1 から 4 のいずれか一項に記載の方法。

【請求項 6】

前記第 1 の参加者が前記第 1 の署名を生成したことを証明するために、前記第 1 の外部データ項目と、前記それぞれのデータ項目に対応する前記取得されたそれぞれの公開鍵とを検証当事者が利用できるようにすることを含む、請求項 5 に記載の方法。

【請求項 7】

前記第 1 のメッセージを取得することは、前記第 1 のメッセージを生成することを含み、前記方法は、前記第 1 のメッセージを前記検証当事者が利用できるようにすることを含む、請求項 1 から 6 のいずれか一項に記載の方法。

【請求項 8】

第 2 のメッセージを取得することと、

少なくとも前記第 2 のメッセージと前記第 1 の参加者の主秘密鍵とに基づいて第 2 の署

10

20

30

40

50

名を生成することと

を含み、前記外部データ項目は前記第 2 の署名を含む、
請求項 1 から 7 のいずれか一項に記載の方法。

【請求項 9】

各参加者は、同じ第 2 のメッセージに基づいて、それぞれの外部データ項目を生成する、請求項 8 に記載の方法。

【請求項 10】

前記第 1 の参加者の前記主秘密鍵は、前記第 1 の参加者のアイデンティティにリンクされた主公開鍵に対応する、請求項 8 または 9 に記載の方法。

【請求項 11】

前記第 1 の共有秘密鍵の前記第 1 のシェアは、秘密分散方式を使用して生成される、請求項 1 から 10 のいずれか一項に記載の方法。

【請求項 12】

前記第 1 の参加者は前記コーディネータであり、前記方法は、
少なくとも前記しきい値数のそれぞれの署名シェアを受信することと、
第 1 の署名構成要素および第 2 の署名構成要素を含む前記第 1 の署名を生成することと
を含み、前記第 1 の署名構成要素は、前記エフェメラル公開鍵に基づいて生成され、前記第 2 の署名構成要素は、少なくとも前記しきい値数のそれぞれの署名シェアに基づいて生成される、請求項 1 から 11 のいずれか一項に記載の方法。

【請求項 13】

前記第 1 のデータ項目は、ランダムソルト値に基づいて生成される、請求項 1 から 12 のいずれか一項に記載の方法。

【請求項 14】

前記方法は、前記ランダムソルト値の知識を証明するためのゼロ知識証明を前記検証当事者に提供することを含む、請求項 13 に記載の方法。

【請求項 15】

前記ランダムソルト値は秘密鍵であり、前記方法は、
第 3 のメッセージを取得することと、
少なくとも前記ランダムソルト値と前記第 3 のメッセージとに基づいて第 3 の署名を生成することと、

前記ランダムソルト値に対応する前記公開鍵を使用して検証されたときに、前記第 3 の署名が前記第 3 のメッセージの有効な署名であることを証明するために、前記第 3 の署名と、前記第 3 のメッセージと、前記ランダムソルト値に対応する公開鍵とを前記検証当事者が利用できるようにすることと

を含む、請求項 13 または 14 に記載の方法。

【請求項 16】

前記第 3 のメッセージは前記第 2 のメッセージを含む、請求項 15 に記載の方法。

【請求項 17】

ハッシュツリーのルートハッシュを生成することであって、前記それぞれのデータ項目に対応する各それぞれの公開鍵が、前記ハッシュツリーのそれぞれのリーフハッシュを生成するためにハッシュされる、ことと

前記ハッシュツリーの前記ルートを前記参加者のうちの 1 つまたは複数および／または前記検証当事者に送信することと

請求項 1 から 16 のいずれか一項に記載の方法。

【請求項 18】

前記第 1 のデータ項目に対応する前記第 1 の公開鍵が前記ハッシュツリーの要素であることを検証するために、ハッシュプルーフを前記検証当事者に送信することを含む、請求項 15 に記載の方法。

【請求項 19】

前記第 1 のメッセージは、ブロックチェーンランザクションの少なくとも一部を含む

10

20

30

40

50

、請求項 1 から 18 のいずれか一項に記載の方法。

【請求項 20】

前記ブロックチェーントランザクションは前記ハッシュルートを含む、請求項 18 および 19 に記載の方法。

【請求項 21】

前記ブロックチェーントランザクションは、前記第 1 の署名を含む、請求項 19 または 20 に記載の方法。

【請求項 22】

前記第 1 のメッセージを前記検証当事者が利用できるようにすることは、前記ブロックチェーントランザクションを前記ブロックチェーンネットワークに送信することを含む、請求項 19 から 21 のいずれか一項に記載の方法。

【請求項 23】

前記外部データ項目の前記ハッシュは、前記外部データ項目のダブルハッシュである、請求項 1 から 22 のいずれか一項に記載の方法。

【請求項 24】

デジタル署名が第 1 の参加者によって部分的に生成されたことを検証するコンピュータ実装方法であって、前記方法は、検証当事者によって実行され、

第 1 の署名構成要素および第 2 の署名構成要素を含む第 1 の署名を取得することと、

前記第 1 の参加者からの候補の第 1 の外部データ項目と、それぞれのデータ項目に対応する 1 つまたは複数のそれぞれの公開鍵とを、他の参加者ごとに 1 つずつ取得することと

、前記候補の第 1 の外部データ項目のハッシュに基づいて候補の公開鍵を生成することと

、前記候補の公開鍵と、前記取得された 1 つまたは複数の公開鍵とに基づいて、候補のエフェメラル公開鍵を生成することと、

前記候補のエフェメラル公開鍵に基づいて候補の第 1 の署名構成要素を生成することと

、前記候補の第 1 の署名構成要素が前記第 1 の署名構成要素に対応するかどうかに基づいて、前記第 1 の署名が前記第 1 の参加者によって部分的に生成されたことを検証することと

を含む方法。

【請求項 25】

前記第 1 の署名は第 1 のメッセージに署名し、前記方法は、

前記第 1 の署名を生成するために使用された共有秘密鍵に対応する共有公開鍵を取得することと、

前記第 1 の公開鍵を使用して検証されたときに、前記第 1 の署名が前記第 1 のメッセージの有効な署名であることを検証することと

請求項 24 に記載の方法。

【請求項 26】

前記候補の外部データ項目は第 2 の署名である、請求項 24 または 25 に記載の方法。

【請求項 27】

前記第 2 の署名は第 2 のメッセージに署名し、前記方法は、

前記第 2 の署名を生成するために使用された秘密鍵に対応する第 2 の公開鍵を取得することと、

前記第 2 の公開鍵を使用して検証されたときに、前記第 2 の署名が前記第 2 のメッセージの有効な署名であることを検証することと

を含む、請求項 24 から 26 のいずれか一項に記載の方法。

【請求項 28】

前記候補の公開鍵は、ランダムソルト値に基づいて生成され、前記方法は、

前記ランダムソルト値の知識を証明するためのゼロ知識証明を受信することと、

10

20

30

40

50

前記第 1 の参加者が前記ランダムソルト値の知識を有することを検証することを含む、請求項 24 または 27 に記載の方法。

【請求項 29】

前記候補の公開鍵は、ランダムソルト値に基づいて生成され、前記方法は、第 3 のメッセージを取得することと、第 3 の署名を取得することと、

前記ランダムソルト値に対応する公開鍵を使用して検証されたときに、前記第 3 の署名が前記第 3 のメッセージの有効な署名であることを検証することを含む、請求項 24 から 28 のいずれか一項に記載の方法。

【請求項 30】

ハッシュブーフおよびハッシュルートを取得することであって、前記第 1 のメッセージは前記ハッシュルートを含む、ことと、

前記ハッシュブーフに基づいて、前記候補の第 1 のエフェメラル公開鍵シェアが、前記ハッシュルートを含むハッシュツリーの要素であることを検証することを含む、請求項 24 から 29 のいずれか一項に記載の方法。

【請求項 31】

前記第 1 の署名は、前記第 1 の当事者から受信される、請求項 24 から 30 のいずれか一項に記載の方法。

【請求項 32】

前記第 1 のメッセージは、前記第 1 の参加者から受信される、請求項 24 から 31 のいずれか一項に記載の方法。

【請求項 33】

前記第 1 のメッセージは、前記検証当事者によって生成される、請求項 24 から 31 のいずれか一項に記載の方法。

【請求項 34】

前記第 1 のメッセージは、ブロックチェーントランザクションの少なくとも一部を含む、請求項 24 から 33 のいずれか一項に記載の方法。

【請求項 35】

前記第 1 のメッセージを取得することは、前記ブロックチェーンから前記ブロックチェーントランザクションを取得することを含む、請求項 34 に記載の方法。

【請求項 36】

前記第 1 の署名を取得することは、前記ブロックチェーントランザクションから前記第 1 の署名を抽出することを含む、請求項 35 に記載の方法。

【請求項 37】

前記ブロックチェーントランザクションの入力は、前記第 1 の署名を含み、前記方法は、

前記ブロックチェーントランザクションの前記入力によって参照される前のブロックチェーントランザクションの出力が署名検証スクリプトを含むことを検証することを含む、請求項 33 から 36 のいずれか一項に記載の方法。

【請求項 38】

コンピュータ機器であって、

1 つまたは複数のメモリユニットを備えるメモリと、

1 つまたは複数の処理ユニットを備える処理装置と

を備え、前記メモリは、前記処理装置上で実行されるように構成されたコードを記憶し、前記コードは、前記処理装置上にあるときに、請求項 1 から 37 のいずれか一項に記載の方法を実行するように構成される、

コンピュータ機器。

【請求項 39】

コンピュータ可読ストレージ上に具現化されたコンピュータプログラムであって、1 つまたは複数のプロセッサ上で実行されると、請求項 1 から 37 のいずれか一項に記載の方

10

20

30

40

50

法を実行するように構成されたコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、デジタル署名のシェア (share) を生成する方法に関する。特に、本方法は、外部データがデジタル署名のシェア内に埋め込まれることを可能にする。

【背景技術】

【0002】

一般に、共有シークレットは、参加者のグループの間で分散されるデータ項目を共有するために使用され得る。各参加者は、シークレットの異なるシェアを有する。通常、シークレットは、特定の数 (「しきい値」と呼ばれる) の参加者が、例えば、シークレットを計算するために一緒に組み合わせられるように、それぞれのシェアを利用できるようにしたときにのみ、再構築されることができる。

【0003】

公開鍵暗号法は、鍵のペア、すなわち、秘密鍵の所有者のみに知られている秘密鍵と、対応する秘密鍵に基づいて生成され、秘密鍵のセキュリティを損なうことなく配布される公開鍵とを使用する暗号システムの一つである。

【0004】

公開鍵暗号法は、送信者が受信者の公開鍵 (すなわち、受信者のみに知られている秘密鍵に対応する公開鍵) を使用してメッセージを暗号化することを可能にする。そのため、暗号化されたメッセージは、受信者の秘密鍵を使用してのみ復号することができる。

【0005】

同様に、送信者は、例えば、メッセージが送信者によって送信されていることを証明するために、および／または、送信者がメッセージに同意することを示すために、自身の秘密鍵を使用してメッセージに署名することができる。署名者 (すなわち、署名を生成する当事者) は、自身の秘密鍵を使用して、メッセージに基づいてデジタル署名を作成する。メッセージに基づいてデジタル署名を作成することは、メッセージおよび秘密鍵の両方に基づいて署名を生成する関数に、メッセージおよび秘密鍵を供給することを意味する。署名は、メッセージに追加される (例えば、タグ付けされる) か、または他の方法でメッセージに関連付けられる。署名者の対応する公開鍵を有する者であれば、同じメッセージおよびメッセージ上のデジタル署名を使用して、署名が有効に作成されたかどうか、すなわち、署名が実際に署名者の秘密鍵を使用して作成されたかどうかを検証することができる。デジタル署名は、メッセージの真正性を保証するだけでなく、メッセージの完全性および否認防止も保証する。すなわち、デジタル署名は、メッセージが署名で署名されてから変更されていないこと、および署名の作成者が署名を作成したことを将来否定することができないことを証明するために使用することができる。

【0006】

デジタル署名方式は、典型的に、3つの手順、すなわちアルゴリズムを含む。鍵生成アルゴリズムは、ランダムな秘密鍵および対応する公開鍵を生成するために使用される。署名アルゴリズムは、メッセージと秘密鍵とに基づいて署名を生成するために使用される。検証アルゴリズムは、公開鍵およびメッセージが与えられた場合に、署名が、対応する秘密鍵を使用して、および、署名アルゴリズムにしたがって生成されたかどうかを検証するために使用される。

【0007】

共有シークレットの一般的な使用は、秘密鍵－公開鍵ペアの共有秘密鍵としての使用である。すなわち、秘密鍵は、単一の参加者では秘密鍵を入手できないように、参加者のグループの間で分散され得る。したがって、単一の参加者がメッセージの有効な署名を生成することはできない。代わりに、署名が生成されるためには、参加者の一部または全部と一緒に秘密鍵を生成しなければならない。

【0008】

10

20

30

40

50

署名を生成するために参加者が自身の秘密鍵シェアを共有する代わりに、参加者は、しきい値署名方式を使用してもよい。しきい値署名方式では、グループ内のしきい値数の参加者は、秘密鍵を参加者が利用できるようにすることなく、共有秘密鍵の個々のシェアを使用して、メッセージに基づいてデジタル署名を作成することができる。ここで、デジタル署名は、署名されるべきメッセージに基づいて生成される署名である。そのような方式では、署名は、しきい値数の参加者が、メッセージに対する署名を生成することに同意する場合にのみ作成することができる。より少ない数の参加者を使用して署名を生成しようとしても、有効な署名は生成されない。したがって、グループによる有効な署名（すなわち、メッセージおよび共有秘密鍵を使用して生成された署名）は、証明可能に、しきい値数の人が署名を生成することに同意していることを意味する。これはまた、秘密鍵を用いて署名を偽造するためには、任意の敵対者がその秘密鍵のしきい値数のシェアを取得する必要があることを意味する。

10

【発明の概要】

【0009】

上述したように、共有シークレットは、共有秘密鍵であり得、参加者のグループの各々は、共有秘密鍵のそれぞれのシェアを有する。しきい値数の参加者は、共有秘密鍵を実際に生成することなく、共有秘密鍵を使用して有効な署名と一緒に生成し得る。公開鍵は、署名を妥当性確認するために、すなわち、少なくともしきい値数の参加者が署名を生成することに同意したことを検証するために、第三者が利用できるようにされ得る。これらの場合、参加者は、自身が署名の生成に寄与したことを証明することができることが望ましい。

20

【0010】

本明細書に開示される一態様によれば、デジタル署名のシェアを生成するコンピュータ実装方法であって、参加者のグループの各参加者は、第1の共有秘密鍵のそれぞれのシェアを有し、方法は、グループの第1の参加者によって実行され、第1のメッセージを取得することと、少なくとも第1の外部データ項目のハッシュに基づいて第1のデータ項目を生成することと、エフェメラル秘密鍵 (ephemeral private key) の第1のエフェメラル秘密鍵シェアを生成することであって、第1のエフェメラル秘密鍵シェアは、第1のデータ項目と、各他の参加者によって生成されたそれぞれのデータ項目とに基づいて生成される、ことと、エフェメラル秘密鍵に対応するエフェメラル公開鍵 (ephemeral public key) を生成することと、第1のメッセージと、第1のエフェメラル秘密鍵シェアと、第1の共有秘密鍵の第1のシェアと、エフェメラル公開鍵とに基づいて、第1の署名シェア (signature share) を生成することと、少なくともしきい値数のそれぞれの署名シェアに基づいて第1の署名を生成するために、第1の署名シェアをコーディネータが利用できるようにすることを含む方法が提供される。

30

【0011】

本明細書で開示される別の態様によれば、デジタル署名が第1の参加者によって部分的に生成されたことを検証するコンピュータ実装方法であって、方法は、検証当事者によって実行され、第1の署名構成要素 (signature component) および第2の署名構成要素を含む第1の署名を取得することと、第1の参加者からの候補の第1の外部データ項目と、それぞれのデータ項目に対応する1つまたは複数のそれぞれの公開鍵とを、他の参加者ごとに1つずつ取得することと、候補の第1の外部データ項目のハッシュに基づいて候補の公開鍵を生成することと、候補の公開鍵と、取得された1つまたは複数の公開鍵とに基づいて、候補のエフェメラル公開鍵を生成することと、候補のエフェメラル公開鍵に基づいて候補の第1の署名構成要素を生成することと、候補の第1の署名構成要素が第1の署名構成要素に対応するかどうかに基づいて、第1の署名が第1の参加者によって部分的に生成されたことを検証することを含む方法が提供される。

40

【0012】

第1の参加者は、メッセージに署名するためのデジタル署名のシェアを生成する。ブロックチェーンの文脈では、メッセージは、トランザクションであり得、例えば、署名は、

50

前のトランザクションの出力をロック解除するためにトランザクションの入力に含まれ得る。一般に、メッセージは、任意の形態のメッセージ、例えば文書であり得、必ずしもブロックチェーンに関連する必要はない。署名シェアは、エフェメラル秘密鍵シェア、例えば、1回使用の秘密鍵のシェアに少なくとも部分的に基づいて生成される。エフェメラル秘密鍵シェアは、外部情報、すなわち「外部データ項目」に少なくとも部分的に基づいて生成される。外部データ項目は、第1の参加者の識別子、例えば、名前、住所、電話番号、国民保険番号、パスポート番号、公開鍵などを含み得、および／またはそれに基づいて生成され得る。いくつかの例では、外部データ項目は、第1の参加者によって生成されたデジタル署名である。第1の参加者は、十分な他の署名シェアとともに完全な署名を生成するために、署名シェアをコーディネータが利用できるようにする。コーディネータは、

10

【0013】

一旦生成されると、署名は、共有秘密鍵に対応する公開鍵を使用して検証することができる。しかしながら、それでは、第1の参加者が署名のシェアを生成したことを第1の参加者が検証当事者に証明することはできない。むしろ、第1の参加者のみが、署名シェアを生成するために使用される外部データ項目を知っているため、第1の参加者は、外部データ項目を明らかにし、検証当事者が署名の少なくとも一部（すなわち、「第1の署名構成要素」）を再構築することを可能にすることができる。すなわち、検証当事者は、外部データ項目に基づいて候補の第1の署名構成要素を生成する。再構築された第1の署名構成要素（すなわち、候補の第1の署名構成要素）が第1の署名構成要素と一致すると、検証当事者は、第1の参加者が実際に署名に寄与したことを確信することができる。

20

【0014】

本発明は、外部情報が署名に組み込まれること、すなわち、署名内に埋め込まれることを可能にする。外部情報は、第1の参加者によって提供されない限り、検証当事者には知られない。具体的には、外部情報は、署名を導出するために使用される。例えば、第1の参加者は、第1の参加者のアイデンティティにリンクされた公開鍵を、異なる秘密鍵（すなわち、埋め込まれた公開鍵に対応しない共有秘密鍵）を使用して作成された署名に埋め込み得る。これにより、第1の参加者は、自身が署名のシェアを生成したことを証明することができる。

【図面の簡単な説明】

30

【0015】

本開示の実施形態の理解を助け、そのような実施形態がどのように実施され得るかを示すために、単なる例として添付の図面を参照する。

【図1】ブロックチェーンを実装するためのシステムの概略ブロック図である。

【図2】ブロックチェーンに記録され得るトランザクションのいくつかの例を概略的に示す。

【図3A】クライアントアプリケーションの概略ブロック図である。

【図3B】図3Aのクライアントアプリケーションによって提示され得る例示的なユーザーインターフェースの概略モックアップである。

【図4】本発明の実施形態を実装するための例示的なシステムの概略ブロック図である。

40

【図5】本発明のいくつかの実施形態による、デジタル署名シェアを生成するための例示的な方法を示すフローチャートである。

【図6】本発明のいくつかの実施形態による、当事者がデジタル署名シェアを生成したことを検証するための例示的な方法を示すフローチャートである。

【図7】本発明のいくつかの実施形態にしたがって生成された例示的なマークルツリーを示す。

【発明を実施するための形態】

【0016】

例示的なシステムの概要

図1は、ブロックチェーン150を実装するための例示的なシステム100を示す。シ

50

ステム 100 は、典型的にはインターネットなどの広域インターネットワークであるパケット交換ネットワーク 101 で構成され得る。パケット交換ネットワーク 101 は、パケット交換ネットワーク 101 内にピアツーピア (P2P) ネットワーク 106 を形成するように構成され得る複数のブロックチェーンノード 104 を含む。図示されていないが、ブロックチェーンノード 104 は、ほぼ完全なグラフとして構成され得る。したがって、各ブロックチェーンノード 104 は、他のブロックチェーンノード 104 に高度に接続される。

【0017】

各ブロックチェーンノード 104 は、ピアのコンピュータ機器を含み、ノード 104 の異なるものは、異なるピアに属する。各ブロックチェーンノード 104 は、1 つまたは複数のプロセッサ、例えば、1 つまたは複数の中央処理装置 (CPU)、アクセラレータプロセッサ、特定用途向けプロセッサおよび／またはフィールドプログラマブルゲートアレイ (FPGA)、ならびに特定用途向け集積回路 (ASIC) などの他の機器を含む処理装置を備える。各ノードはまた、メモリ、すなわち、1 つまたは複数の非一時的コンピュータ可読媒体の形態のコンピュータ可読ストレージを備える。メモリは、1 つまたは複数のメモリ媒体、例えば、ハードディスクなどの磁気媒体、ソリッドステートドライブ (SSD)、フラッシュメモリもしくは EEPROM などの電子媒体、および／または光ディスクドライブなどの光学媒体を採用する 1 つまたは複数のメモリユニットを備え得る。

【0018】

ブロックチェーン 150 は、データブロック 151 のチェーンを含み、ブロックチェーン 150 のそれぞれのコピーは、分散型またはブロックチェーンネットワーク 106 内の複数のブロックチェーンノード 104 の各々で維持される。上述したように、ブロックチェーン 150 のコピーを維持することは、ブロックチェーン 150 を完全に記憶することを必ずしも意味しない。代わりに、ブロックチェーン 150 は、各ブロックチェーンノード 150 が各ブロック 151 のブロックヘッダ (後述する) を記憶している限り、データがブルーニングされ得る。チェーン内の各ブロック 151 は、1 つまたは複数のトランザクション 152 を含み、この文脈におけるトランザクションは、データ構造の一種を指す。データ構造の性質は、トランザクションモデルまたは方式の一部として使用されるトランザクションプロトコルのタイプに依存する。所与のブロックチェーンは、全体を通して 1 つの特定のトランザクションプロトコルを使用する。1 つの一般的なタイプのトランザクションプロトコルでは、各トランザクション 152 のデータ構造は、少なくとも 1 つの入力および少なくとも 1 つの出力を含む。各出力は、特性としてデジタル資産の量を表す額を指定し、その例は、出力が暗号的にロックされている (ロック解除され、それによって償還または使用されるためにはそのユーザの署名または他のソリューションを必要とする) ユーザ 103 である。各入力は、先行するトランザクション 152 の出力を指し示し、それによってトランザクションをリンクする。

【0019】

各ブロック 151 はまた、ブロック 151 への順番を定義するために、チェーン内の前に作成されたブロック 151 を指し示すブロックポインタ 155 を含む。各トランザクション 152 (コインベーストランザクション以外) は、トランザクションのシーケンスへの順序を定義するために、前のトランザクションへ戻るポインタを含む (注意: トランザクション 152 のシーケンスは分岐することが可能である)。ブロック 151 のチェーンは、チェーン内の最初のブロックであった発生ブロック (Gb: genesis block) 153 までずっと戻る。チェーン 150 内の早期にある 1 つまたは複数の元のトランザクション 152 は、先行するトランザクションではなく発生ブロック 153 を指し示していた。

【0020】

ブロックチェーンノード 104 の各々は、トランザクション 152 を他のブロックチェーンノード 104 にフォワードし、それによってトランザクション 152 をネットワーク 106 全体に伝搬させるように構成される。各ブロックチェーンノード 104 は、ブロック 151 を作成し、同じブロックチェーン 150 のそれぞれのコピーをそれらのそれぞれ

10

20

30

40

50

のメモリに記憶するように構成される。各ブロックチェーンノード104はまた、ブロック151に組み込まれるのを待っているトランザクション152の順序付きセット（またはプール）154を維持する。順序付きプール154は、「メムプール（mempool）」と呼ばれることが多い。本明細書におけるこの用語は、任意の特定のブロックチェーン、プロトコル、またはモデルに限定することを意図していない。これは、ノード104が有効であるとして受け入れたトランザクションの順序付きセットを指し、それに対して、ノード104は、同じ出力を使用しようとする他のトランザクションを受け入れないように義務付けられている。

【0021】

所与の現在のトランザクション152jにおいて、その入力（または各入力）は、トランザクションのシーケンスにおける先行するトランザクション152iの出力を参照するポインタを含み、この出力が現在のトランザクション152jにおいて償還または「使用」されるべきであることを指定する。一般に、先行するトランザクションは、順序付きセット154または任意のブロック151内の任意のトランザクションであり得る。先行するトランザクション152iは、現在のトランザクションが有効となるためには存在および妥当性確認される必要があるが、先行するトランザクション152iは、現在のトランザクション152jが作成されるときまたはネットワーク106に送信されるときに必ずしも存在する必要はない。したがって、本明細書における「先行する（preceding）」は、ポインタによってリンクされた論理シーケンスにおける先行するものを指し、必ずしも時間シーケンスにおける作成または送信の時間を指すものではなく、したがって、トランザクション152i、152jが順不同に作成または送信されることを必ずしも除外するものではない（オーファントランザクションに関する以下の説明を参照）。先行するトランザクション152iは、同様に、先のトランザクション（antecedent transaction）または先行したトランザクション（predecessor transaction）とも呼ばれる。

【0022】

現在のトランザクション152jの入力はまた、入力認可、例えば、先行するトランザクション152iの出力がロックされている先のユーザ103aの署名を含む。次に、現在のトランザクション152jの出力は、新しいユーザまたはエンティティ103bに暗号的にロックされ得る。したがって、現在のトランザクション152jは、先行するトランザクション152iの入力において定義された額を、現在のトランザクション152jの出力において定義されたように、新しいユーザまたはエンティティ103bに転送することができる。場合によっては、トランザクション152は、複数のユーザまたはエンティティ（そのうちの1つは残り（change））を与えるために元のユーザまたはエンティティ103aであり得る）間で入力額を分割するために複数の出力を有し得る。場合によっては、トランザクションはまた、1つまたは複数の先行するトランザクションの複数の出力からの額をまとめ、現在のトランザクションの1つまたは複数の出力に再分配するために複数の入力を有することができる。

【0023】

ビットコインなどの出力ベースのトランザクションプロトコルによれば、個々のユーザまたは組織などの当事者103が（手動でまたは当事者によって採用される自動プロセスによって）新しいトランザクション152jを制定することを望むとき、制定を行う当事者は、新しいトランザクションをそのコンピュータ端末102から受信者に送信する。制定を行う当事者または受信者は、最終的に、このトランザクションをネットワーク106のブロックチェーンノード104の1つまたは複数（これは、現在では、典型的にはサーバまたはデータセンタであるが、原則として他のユーザ端末であってもよい）に送信する。新しいトランザクション152jを制定する当事者103がトランザクションをブロックチェーンノード104の1つまたは複数に直接送信し、いくつかの例では、受信者に送信しないことも除外されない。トランザクションを受信するブロックチェーンノード104は、ブロックチェーンノード104の各々で適用されるブロックチェーンノードプロトコルにしたがって、トランザクションが有効であるかどうかをチェックする。ブロックチ

チェーンノードプロトコルは、典型的には、新しいトランザクション152j内の暗号署名が、トランザクション152の順序付きシーケンス内で前のトランザクション152iに依存する予想される署名と一致することをチェックするようにブロックチェーンノード104に求める。そのような出力ベースのトランザクションプロトコルでは、これは、新しいトランザクション152jの入力に含まれる当事者103の暗号署名または他の認可が、新しいトランザクションが割り当てる先行するトランザクション152iの出力において定義される条件と一致することをチェックすることを含み得、この条件は、典型的には、新しいトランザクション152jの入力における暗号署名または他の認可が、新しいトランザクションの入力がリンクされている前のトランザクション152iの出力をロック解除することを少なくともチェックすることを含む。条件は、先行するトランザクション152iの出力に含まれるスクリプトによって少なくとも部分的に定義され得る。代替的に、単にブロックチェーンノードプロトコルのみによって固定されてもよく、またはこれらの組合せによるものであってもよい。いずれにしても、新しいトランザクション152jが有効である場合、ブロックチェーンノード104は、それをブロックチェーンネットワーク106内の1つまたは複数の他のブロックチェーンノード104にフォワードする。これらの他のブロックチェーンノード104は、同じブロックチェーンノードプロトコルにしたがって同じテストを適用し、そして、新しいトランザクション152jを1つまたは複数のさらなるノード104にフォワードし、以下同様である。このようにして、新しいトランザクションはブロックチェーンノード104のネットワーク全体に伝搬される。

10

20

【0024】

出力ベースのモデルでは、所与の出力（例えば、UTXO）が割り当てられる（例えば、使用される）かどうかの定義は、それがブロックチェーンノードプロトコルにしたがって別の前方のトランザクション152jの入力によって有効に償還されたかどうかである。トランザクションが有効であるための別の条件は、それが償還しようとする先行するトランザクション152iの出力が、別のトランザクションによってまだ償還されていないことである。この場合も同様に、有効ではない場合、トランザクション152jは、（無効としてフラグ付けされ、警告のために伝搬されない限り）伝搬されることも、ブロックチェーン150内に記録されることもない。これは、トランザクタ（transactor）が同じトランザクションの出力を複数回割り当てようとする二重支出を防止する。一方、アカウントベースのモデルは、アカウント残高を維持することによって二重支出を防止する。ここでも、トランザクション順序が定義されているので、アカウント残高は常に単一の定義された状態にある。

30

【0025】

トランザクションを妥当性確認することに加えて、ブロックチェーンノード104はまた、「ブルーフオブワーク」によって支持される、一般にマイニングと呼ばれるプロセスにおいてトランザクションのブロックを最初に作成しようと競い合う。ブロックチェーンノード104において、新しいトランザクションが、ブロックチェーン150上に記録されたブロック151内にまだ現れていない有効なトランザクションの順序付きプール154に追加される。次いで、ブロックチェーンノードは、暗号パズルを解こうとすることで、トランザクションの順序付きセット154からトランザクション152の新しい有効なブロック151を組み立てようと競い合う。典型的には、これは、ナンスが保留中のトランザクションの順序付きプール154の表現と連結されハッシュされたときにハッシュの出力が所定の条件を満たすような「ナンス」値を探索することを含む。例えば、所定の条件とは、ハッシュの出力が特定の所定の数の先行ゼロを有することであり得る。これは、ブルーフオブワークパズルの1つの特定のタイプにすぎず、他のタイプが除外されないことに留意されたい。ハッシュ関数の特性は、その入力に対して予測不可能な出力を持つことである。したがって、この探索は、総当たりでしか実行することができないので、パズルを解こうとしている各ブロックチェーンノード104でかなりの量の処理リソースを消費する。

40

50

【0026】

最初にパズルを解いたブロックチェーンノード104は、これをネットワーク106に公表し、後にネットワーク内の他のブロックチェーンノード104によって容易にチェックすることができるその解を証拠として提供する（ハッシュに対する解が与えられると、ハッシュの出力が条件を満たすことをチェックすることは簡単である）。この最初のブロックチェーンノード104は、ブロックを、このブロックを受け入れる他のノードのしきい値コンセンサスに伝搬し、プロトコルルールを実施する。次いで、トランザクションの順序付きセット154は、ブロックチェーンノード104の各々によってブロックチェーン150内に新しいブロック151として記録されるようになる。ブロックポインタ155はまた、チェーン内の前に作成されたブロック151_{n-1}を指し示す新しいブロック151_nに割り当てられる。プルーフオブワークの解を作成するために必要とされる、例えばハッシュの形態の、かなりの量の労力は、ブロックチェーンプロトコルのルールに従うという最初のノード104の意図を示す。そのようなルールは、別名二重支出としても知られている、前に妥当性確認されたトランザクションと同じ出力の割当てを行う場合、トランザクションを有効として受け入れないことを含む。ブロック151は、一旦作成されると、ブロックチェーンネットワーク106内のブロックチェーンノード104の各々において認識および維持されるので、修正することができない。ブロックポインタ155はまた、ブロック151に順番を付与する。トランザクション152は、ネットワーク106内の各ブロックチェーンノード104において順序付きブロックに記録されるので、これは、トランザクションの不変の公開台帳を提供する。

【0027】

任意の所与の時間にパズルを解こうと競い合う異なるブロックチェーンノード104は、それらがいつ解を探索し始めたかまたはトランザクションが受信された順序に応じて、任意の所与の時間に、まだ公開されていないトランザクションのプール154の異なるスナップショットに基づいてそれを行っていてもよいことに留意されたい。誰がそれぞれのパズルを最初に解いても、どのトランザクション152が次の新しいブロック151_nにどの順序で含まれるかを定義し、公開されていないトランザクションの現在のプール154が更新される。次いで、ブロックチェーンノード104は、新たに定義された、公開されていないトランザクションの順序付きプール154からブロックを作成しようと競い合い続け、以下同様である。2つのブロックチェーンノード104が互いに非常に短い時間内にパズルを解いて、ブロックチェーンの相反する見解がノード104間で伝搬される場合に発生し得る任意の「フォーク」を解決するためのプロトコルも存在する。要するに、フォークのどちらのブランチが最も長く成長しても、確定的なブロックチェーン150となる。同じトランザクションが両方のフォークに現れるので、これがネットワークのユーザまたはエージェントに影響を与えないことに留意されたい。

【0028】

ビットコインブロックチェーン（およびほとんどの他のブロックチェーン）によれば、新しいブロック104の構築に成功したノードには、（あるエージェントまたはユーザから別のエージェントまたはユーザにある額のデジタル資産を転送するエージェント間またはユーザ間のトランザクションとは対照的に）追加の定義された量のデジタル資産を分配する新しい特別な種類のトランザクションにおいて、受け入れられた追加の額のデジタル資産を新たに割り当てる能力が与えられる。この特別なタイプのトランザクションは、通常、「コインベーストランザクション」と呼ばれるが、「開始トランザクション（initiation transaction）」または「生成トランザクション（generation transaction）」と呼ばれることもある。これは典型的に、新しいブロック151_nの最初のトランザクションを形成する。プルーフオブワークは、新しいブロックを構築するノードが、この特別なトランザクションが後に償還されることを可能にするプロトコルルールに従うという意図を示す。ブロックチェーンプロトコルルールは、この特別なトランザクションが償還され得る前に、満期期間、例えば100個のブロックを必要とし得る。多くの場合、通常の（非生成）トランザクション152はまた、そのトランザクションが公開されたブ

ック151nを作成したブロックチェーンノード104にさらに報酬を与えるために、その出力のうちの1つにおいて追加のトランザクション手数料を指定する。この手数料は通常「トランザクション手数料」と呼ばれ、以下に説明する。

【0029】

トランザクション妥当性確認および公開に関与するリソースに起因して、典型的には、ブロックチェーンノード104の少なくとも各々は、1つまたは複数の物理サーバユニットを含むサーバの形態をとるか、さらにはデータセンタ全体の形態をとる。しかしながら、原則として、任意の所与のブロックチェーンノード104は、ユーザ端末または互いにネットワーク化されたユーザ端末のグループの形態をとることができる。

【0030】

各ブロックチェーンノード104のメモリは、そのそれぞれの1つまたは複数の役割を実行し、ブロックチェーンノードプロトコルにしたがってトランザクション152を処理するために、ブロックチェーンノード104の処理装置上で実行されるように構成されたソフトウェアを記憶する。本明細書においてブロックチェーンノード104に帰する任意のアクションは、それぞれのコンピュータ機器の処理装置上で実行されるソフトウェアによって実行され得ることが理解されよう。ノードソフトウェアは、アプリケーション層、またはオペレーティングシステム層もしくはプロトコル層などの下位層、またはこれらの任意の組合せにおける1つまたは複数のアプリケーションにおいて実装され得る。

【0031】

消費ユーザの役割を果たす複数の当事者103の各々のコンピュータ機器102もネットワーク101に接続されている。これらのユーザは、ブロックチェーンネットワーク106と対話し得るが、トランザクションの妥当性確認にもブロックの構築にも参加しない。これらのユーザまたはエージェント103のうちのいくつかは、トランザクションの送信者および受信者として動作し得る。他のユーザは、必ずしも送信者または受信者として動作することなくブロックチェーン150と対話し得る。例えば、いくつかの当事者は、（例えば、ブロックチェーンノード104からブロックチェーンのコピーを取得した）ブロックチェーン150のコピーを記憶するストレージエンティティとして動作し得る。

【0032】

当事者103のうちのいくつかまたはすべては、異なるネットワーク、例えば、ブロックチェーンネットワーク106の上にオーバーレイされたネットワークの一部として接続され得る。ブロックチェーンネットワークのユーザ（「クライアント」と呼ばれることが多い）は、ブロックチェーンネットワーク106を含むシステムの一部であるといえるが、これらのユーザは、ブロックチェーンノードに求められる役割を果たさないで、ブロックチェーンノード104ではない。代わりに、各当事者103はブロックチェーンネットワーク106と対話してもよく、ブロックチェーンノード106に接続する（すなわち通信する）ことでブロックチェーン150を利用し得る。2つの当事者103およびそれらのそれぞれの機器102、すなわち、第1の当事者103aおよびそのそれぞれのコンピュータ機器102a、ならびに第2の当事者103bおよびそのそれぞれのコンピュータ機器102bは、例示の目的で示されている。そのような当事者103およびそれらのそれぞれのコンピュータ機器102ははるかに多く存在し、システム100に参加し得るが、便宜上、それらは図示されていないことが理解されよう。各当事者103は、個人または組織であり得る。純粋に例示として、第1の当事者103aは、本明細書ではアリスと呼ばれ、第2の当事者103bはボブと呼ばれるが、これは限定的なものではなく、本明細書におけるアリスまたはボブへのいかなる言及も、それぞれ「第1の当事者」および「第2の当事者」と置き換えられ得ることが理解されよう。

【0033】

各当事者103のコンピュータ機器102は、1つまたは複数のプロセッサ、例えば、1つまたは複数のCPU、GPU、他のアクセラレータプロセッサ、特定用途向けプロセッサ、および／またはFPGAを含むそれぞれの処理装置を備える。各当事者103のコンピュータ機器102は、メモリ、すなわち、1つまたは複数の非一時的コンピュータ可

10

20

30

40

50

読媒体の形態のコンピュータ可読ストレージをさらに備える。このメモリは、1つまたは複数のメモリ媒体、例えば、ハードディスクなどの磁気媒体、SSD、フラッシュメモリもしくはEEPROMなどの電子媒体、および／または光ディスクドライブなどの光学媒体を採用する1つまたは複数のメモリユニットを備え得る。各当事者103のコンピュータ機器102上のメモリは、処理装置上で実行されるように構成された少なくとも1つのクライアントアプリケーション105のそれぞれのインスタンスを含むソフトウェアを記憶する。本明細書において所与の当事者103に帰する任意のアクションは、それぞれのコンピュータ機器102の処理装置上で実行されるソフトウェアを使用して実行され得ることが理解されよう。各当事者103のコンピュータ機器102は、少なくとも1つのユーザ端末、例えば、デスクトップもしくはラップトップコンピュータ、タブレット、スマートフォン、またはスマートウォッチなどのウェアラブルデバイスを含む。所与の当事者103のコンピュータ機器102はまた、ユーザ端末を介してアクセスされるクラウドコンピューティングリソースなどの1つまたは複数の他のネットワーク化されたリソースを含み得る。

10

20

30

40

50

【0034】

クライアントアプリケーション105は、最初に、1つまたは複数の適切なコンピュータ可読ストレージ上で任意の所与の当事者103のコンピュータ機器102に提供され得、例えば、サーバからダウンロードされ得るか、またはリムーバブルSSD、フラッシュメモリキー、リムーバブルEEPROM、リムーバブル磁気ディスクドライブ、磁気フロッピーディスクもしくはテープ、CDもしくはDVD ROMなどの光ディスク、またはリムーバブル光学ドライブなどのリムーバブル記憶デバイス上で提供され得る。

【0035】

クライアントアプリケーション105は、少なくとも「ウォレット」機能を備える。これは2つの主要な機能を有する。これらのうちの1つは、それぞれの当事者103が、トランザクション152を作成し、認可し（例えば署名し）、1つまたは複数のビットコインノード104に送信することを可能にして、トランザクション152を、ブロックチェーンノード104のネットワーク全体に伝搬させ、それによってブロックチェーン150に含まれるようにすることである。もう1つは、それぞれの当事者に、その当事者が現在所有しているデジタル資産の額を報告することである。出力ベースのシステムでは、この第2の機能は、当該当事者に属するブロックチェーン150全体に散在している様々なトランザクション152の出力において定義された額を照合することを含む。

【0036】

様々なクライアント機能が、所与のクライアントアプリケーション105に統合されるものとして説明され得るが、必ずしもこれに限定されず、代わりに、本明細書で説明される任意のクライアント機能は、例えば、APIを介してインターフェースする、または一方が他方へのプラグインである2つ以上の別個のアプリケーション一式において実装され得ることに留意されたい。より一般的には、クライアント機能は、アプリケーション層もしくはオペレーティングシステムなどの下位層、またはこれらの任意の組合せにおいて実装され得る。以下では、クライアントアプリケーション105に関して説明するが、これに限定されないことが理解されよう。

【0037】

各コンピュータ機器102上のクライアントアプリケーションまたはソフトウェア105のインスタンスは、ネットワーク106のブロックチェーンノード104のうちの少なくとも1つに動作可能に結合される。これにより、クライアント105のウォレット機能はトランザクション152をネットワーク106に送信することができる。クライアント105はまた、それぞれの当事者103が受信者である任意のトランザクションについてブロックチェーン150にクエリを行うためにブロックチェーンノード104にコンタクトすることができる（または、実施形態では、ブロックチェーン150は、部分的にその公開性（public visibility）を通じてトランザクションにおける信頼を提供する公共施設であるので、実際にブロックチェーン150における他の当事者のトランザクション

を検査する)。各コンピュータ機器102上のウォレット機能は、トランザクションプロトコルにしたがってトランザクション152を定式化し、送信するように構成される。上述したように、各ブロックチェーンノード104は、ブロックチェーンノードプロトコルにしたがってトランザクション152を妥当性確認し、トランザクション152をフォワードして、それらをブロックチェーンネットワーク106全体に伝搬するように構成されたソフトウェアを実行する。トランザクションプロトコルおよびノードプロトコルは互に対応し、所与のトランザクションプロトコルは所与のノードプロトコルに従い (go with)、一緒に所与のトランザクションモデルを実装する。ブロックチェーン150内のすべてのトランザクション152に対して同じトランザクションプロトコルが使用される。ネットワーク106内のすべてのノード104によって同じノードプロトコルが使用される。

10

【0038】

所与の当事者103、例えばアリスが、ブロックチェーン150に含まれるべき新しいトランザクション152jを送信することを望むとき、アリスは、関連トランザクションプロトコルにしたがって (アリスのクライアントアプリケーション105内のウォレット機能を使用して) 新しいトランザクションを定式化する。次いで、アリスは、クライアントアプリケーション105から、アリスが接続されている1つまたは複数のブロックチェーンノード104にトランザクション152を送信する。例えば、これは、アリスのコンピュータ102に最良に接続されたブロックチェーンノード104であり得る。任意の所与のブロックチェーンノード104は、新しいトランザクション152jを受信すると、ブロックチェーンノードプロトコルおよびそのそれぞれの役割にしたがってそれを処理する。これは、新たに受信されたトランザクション152jが「有効」であるための特定の条件を満たすかを最初にチェックすることを含み、その例については、以下でより詳細に説明する。いくつかのトランザクションプロトコルでは、妥当性確認のための条件は、トランザクション152に含まれるスクリプトによってトランザクションごとに構成可能であり得る。代替的に、条件は、単にノードプロトコルの組み込み特徴であってもよいし、スクリプトとノードプロトコルとの組合せによって定義されてもよい。

20

【0039】

新たに受信されたトランザクション152jが、有効であるとみなされるためのテストにパスすることを条件として (すなわち、それが「妥当性確認される」ことを条件として)、トランザクション152jを受信する任意のブロックチェーンノード104は、そのブロックチェーンノード104において維持されるトランザクションの順序付きセット154に新たな妥当性確認済みトランザクション152を追加する。さらに、トランザクション152jを受信する任意のブロックチェーンノード104は、妥当性確認済みトランザクション152をネットワーク106内の1つまたは複数の他のブロックチェーンノード104へと前方に伝搬する。各ブロックチェーンノード104は同じプロトコルを適用するので、トランザクション152jが有効であると仮定すると、これは、ネットワーク106全体にわたってすぐに伝搬されることを意味する。

30

【0040】

所与のブロックチェーンノード104において維持される保留中のトランザクションの順序付きプール154に承認されると、そのブロックチェーンノード104は、新しいトランザクション152を含むそれぞれのプール154の最新バージョンに対してプルーフオブワークパズルを解こうと競い始める (他のブロックチェーンノード104が、トランザクションの異なるプール154に基づいてパズルを解こうと試みている可能性があるが、どのノードでも最初に解いたものが、最新のブロック151に含まれるトランザクションのセットを定義することを想起されたい。最終的に、ブロックチェーンノード104は、アリスのトランザクション152jを含む順序付きプール154の一部についてパズルを解くことになる。) 新しいトランザクション152jを含むプール154に対してプルーフオブワークが行われると、それは普遍的にブロックチェーン150内のブロック151のうちの1つの一部となる。各トランザクション152は、先のトランザクションへ戻

40

50

るポインタを含むので、トランザクションの順序も不変的に記録される。

【0041】

異なるブロックチェーンノード104は、最初、所与のトランザクションの異なるインスタンスを受信し得るので、1つのインスタンスが新しいブロック151において公開される（この時点で、公開されたインスタンスが唯一の有効なインスタンスであることにすべてのブロックチェーンノード104が同意している）までは、どのインスタンスが「有効」であるかについて相反する見解を有する。ブロックチェーンノード104が1つのインスタンスを有効として受け入れ、次いで、別のインスタンスがブロックチェーン150に記録されていることを発見した場合、そのブロックチェーンノード104は、これを受け入れなければならない、最初に受け入れたインスタンス（すなわち、ブロック151で公開されていないもの）を破棄する（すなわち、無効として扱う）。

10

【0042】

いくつかのブロックチェーンネットワークによって動作される代替タイプのトランザクションプロトコルは、アカウントベースのトランザクションモデルの一部として、「アカウントベース」プロトコルと呼ばれ得る。アカウントベースの場合、各トランザクションは、過去のトランザクションのシーケンスにおける先行するトランザクションのUTXOを参照することによってではなく、絶対アカウント残高を参照することによって転送されるべき額を定義する。すべてのアカウントの現在の状態は、ブロックチェーンとは別個にそのネットワークのノードによって記憶され、絶えず更新される。そのようなシステムでは、トランザクションは、アカウントの実行中のトランザクションタリー（「ポジション」とも呼ばれる）を使用して順序付けられる。この値は、送信者によってその暗号署名の一部として署名され、トランザクション参照計算の一部としてハッシュされる。加えて、トランザクションにおけるオプションのデータフィールドも署名することができる。このデータフィールドは、例えば、前のトランザクションIDがデータフィールドに含まれている場合、前のトランザクションを指し示し得る。

20

【0043】

UTXOベースのモデル

図2は、例示的なトランザクションプロトコルを示す。これは、UTXOベースのプロトコルの一例である。トランザクション152（「Tx」と略記される）は、ブロックチェーン150の基本的なデータ構造である（各ブロック151は1つまたは複数のトランザクション152を含む）。以下では、出力ベースまたは「UTXO」ベースのプロトコルを参照して説明する。しかしながら、これはすべての可能な実施形態に限定されない。例示的なUTXOベースのプロトコルは、ビットコインを参照して説明されるが、他の例示的なブロックチェーンネットワーク上でも等しく実装され得ることに留意されたい。

30

【0044】

UTXOベースのモデルでは、各トランザクション（「Tx」）152は、1つまたは複数の入力202および1つまたは複数の出力203を含むデータ構造を含む。各出力203は、未使用トランザクション出力（UTXO）を含み得、これは、（UTXOがまだ償還されていない場合）別の新しいトランザクションの入力202のソースとして使用され得る。UTXOは、デジタル資産の額を指定する値を含む。これは、分散型台帳上のトークンの設定数を表す。UTXOはまた、他の情報の中でも、元となるトランザクションのトランザクションIDを含み得る。トランザクションデータ構造は、入力フィールド（複数可）202および出力フィールド（複数可）203のサイズを示すインジケータを含み得るヘッダ201も含み得る。ヘッダ201はまた、トランザクションのIDを含み得る。実施形態では、トランザクションIDは、（トランザクションID自体を除く）トランザクションデータのハッシュであり、ノード104にサブミットされる生のトランザクション152のヘッダ201に記憶される。

40

【0045】

アリス103aが、当該デジタル資産の額をボブ103bに転送するトランザクション152jを作成することを望むとする。図2では、アリスの新しいトランザクション15

50

2 j は「Tx₁」とラベル付けされている。これは、シーケンス内の先行するトランザクション152 i の出力203においてアリスにロックされたデジタル資産の額を取り、これのうちの少なくとも一部をボブに転送する。先行するトランザクション152 i は、図2では「Tx₀」とラベル付けされている。Tx₀およびTx₁は、単なる任意のラベルである。それらは、Tx₀がブロックチェーン151内の最初のトランザクションであること、またはTx₁がプール154内のすぐ次のトランザクションであることを必ずしも意味するものではない。Tx₁は、アリスにロックされた未使用の出力203を依然として有する任意の先行する（すなわち先の）トランザクションを指し示すことができる。

【0046】

先行するトランザクションTx₀は、アリスが新しいトランザクションTx₁を作成した時点では、または少なくともアリスがそれをネットワーク106に送信する時点までには、すでに妥当性確認されブロックチェーン150のブロック151に含まれている可能性がある。それは、その時点でブロック151のうちの1つにすでに含まれていてもよいし、順序付きセット154で依然として待機していてもよく、この場合、すぐに新しいブロック151に含まれることになる。代替的に、Tx₀およびTx₁を作成してネットワーク106と一緒に送信することもできるし、ノードプロトコルが「オーファン」トランザクションのバッファリングを可能にする場合には、Tx₀をTx₁の後に送信することさえもできる。トランザクションのシーケンスの文脈において本明細書で使用される「先行する」および「後続の」という用語は、トランザクション内で指定されているトランザクションポインタ（どのトランザクションがどの他のトランザクションを指し示すかなど）によって定義されるシーケンス内のトランザクションの順序を指す。それらは、同様に、「先行するもの（predecessor）」および「後続するもの（successor）」、または「先の」および「後の」、「親」および「子」などと置き換えられ得る。これは、それらの作成、ネットワーク106への送信、または任意の所与のブロックチェーンノード104への到着の順序を必ずしも意味するものではない。それにもかかわらず、先行するトランザクション（先のトランザクションまたは「親」）を指し示す後続するトランザクション（後のトランザクションまたは「子」）は、親トランザクションが妥当性確認されない限り、妥当性確認されない。親より前にブロックチェーンノード104に到着する子は、オーファンとみなされる。それは、ノードプロトコルおよび／またはノード挙動に応じて、親を待つために特定の時間バッファされるかまたは破棄され得る。

【0047】

先行するトランザクションTx₀の1つまたは複数の出力203のうちの1つは、本明細書ではUTXO₀とラベル付けされた特定のUTXOを含む。各UTXOは、UTXOによって表されるデジタル資産の額を指定する値と、ロックスクリプトとを含み、ロックスクリプトは、後続のトランザクションが妥当性確認され、したがってUTXOが正常に償還されるために、後続のトランザクションの入力202内のロック解除スクリプトが満たさなければならない条件を定義する。典型的には、ロックスクリプトは、その額を特定の当事者（それが含まれるトランザクションの受益者）にロックする。すなわち、ロックスクリプトは、典型的には、後続のトランザクションの入力内のロック解除スクリプトに、先行するトランザクションがロックされる当事者の暗号署名が含まれるという条件を含むロック解除条件を定義する。

【0048】

ロックスクリプト（通称scriptPubKey）は、ノードプロトコルによって認識されるド主固有言語で書かれたコードの一部分である。そのような言語の特定の例は、ブロックチェーンネットワークによって使用される「スクリプト（Script）」（大文字S）と呼ばれる。ロックスクリプトは、トランザクション出力203を使用するためにどの情報が必要とされるか、例えばアリスの署名の必要性、を指定する。ロック解除スクリプトはトランザクションの出力に現れる。ロック解除スクリプト（通称scriptSig）は、ロックスクリプト基準を満たすのに必要な情報を提供するド主固有言語で書かれたコードの一部分である。例えば、それはボブの署名を含み得る。ロック解除スクリプトは、トランザクシ

10

20

30

40

50

ヨンの入力 202 に現れる。

【0049】

つまり、図示の例では、 Tx_0 の出力 203 内の $UTXO_0$ は、 $UTXO_0$ が償還されるために（厳密には、 $UTXO_0$ を償還しようとする後続のトランザクションが有効となるために）アリスの署名 $Sig\ P_A$ を必要とするロックスクリプト $[Checksig\ P_A]$ を含む。 $[Checksig\ P_A]$ は、アリスの公開鍵－秘密鍵ペアの公開鍵 P_A の表現（すなわち、ハッシュ）を含む。 Tx_1 の入力 202 は、（例えば、実施形態ではトランザクション Tx_0 全体のハッシュであるそのトランザクション ID、 $TxID_0$ によって） Tx_1 を指し示すポインタを含む。 Tx_1 の入力 202 は、 Tx_0 の任意の他の可能な出力の中から $UTXO_0$ を識別するために、 Tx_0 内の $UTXO_0$ を識別するインデックスを含む。 Tx_1 の入力 202 は、アリスが鍵ペアのアリスの秘密鍵をデータの所定の部分（暗号では「メッセージ」と呼ばれることもある）に適用することによって作成された、アリスの暗号署名を含むロック解除スクリプト $\langle Sig\ P_A \rangle$ をさらに含む。有効な署名を提供するためにアリスによって署名される必要があるデータ（または「メッセージ」）は、ロックスクリプトによって、またはノードプロトコルによって、またはこれらの組合せによって定義され得る。

10

【0050】

新しいトランザクション Tx_1 がブロックチェーンノード 104 に到着すると、ノードはノードプロトコルを適用する。これは、ロックスクリプトおよびロック解除スクリプトと一緒に実行して、ロック解除スクリプトがロックスクリプトで定義されている条件（この条件は 1 つまたは複数の基準を含み得る）を満たすかどうかをチェックすることを含む。実施形態では、これは 2 つのスクリプトを連結することを含む：

20

$\langle Sig\ P_A \rangle \quad \langle P_A \rangle \quad | \quad | \quad [Checksig\ P_A]$

ここで、「 $|$ 」は連結を表し、「 $\langle \dots \rangle$ 」はデータをスタックに置くことを意味し、「 $[\dots]$ 」はロックスクリプト（この例ではスタックベースの言語）で構成される関数である。同等に、スクリプトは、スクリプトを連結するのではなく、共通スタックを用いて次々に実行され得る。いずれにしても、一緒に実行されるとき、スクリプトは、 Tx_0 の出力内のロックスクリプトに含まれるようなアリスの公開鍵 P_A を使用して、 Tx_1 の入力内のロック解除スクリプトが、データの予想される部分に署名したアリスの署名を含むことを認証する。データの予想される部分自体（「メッセージ」）はまた、この認証を実行するために含まれる必要がある。実施形態では、署名されたデータは、 Tx_1 の全体を含む（つまり、平文のデータの署名された部分を指定する別個の要素は、すでに本質的に存在するので、含まれる必要はない）。

30

【0051】

公開－秘密暗号法による認証の詳細は、当業者によく知られている。基本的に、アリスが自身の秘密鍵を使用してメッセージに署名した場合、アリスの公開鍵および平文のメッセージが与えられると、ノード 104 などの別のエンティティは、メッセージがアリスによって署名されたものに違いないことを認証することができる。署名は、典型的には、メッセージをハッシュし、ハッシュに署名し、これを署名としてメッセージにタグ付けすることを含み、これにより、公開鍵の任意の保持者が署名を認証することができる。したがって、データの特定の部分またはトランザクションの一部などに署名することへの本明細書におけるいかなる参照も、実施形態では、データのその部分またはトランザクションの一部のハッシュに署名することを意味し得ることに留意されたい。

40

【0052】

Tx_1 内のロック解除スクリプトが、 Tx_0 のロックスクリプト内で指定されている 1 つまたは複数の条件を満たす場合（つまり、図示の例では、アリスの署名が Tx_1 内で提供され、認証された場合）、ブロックチェーンノード 104 は、 Tx_1 が有効であるとみなす。これは、ブロックチェーンノード 104 が、保留中のトランザクションの順序付きプール 154 に Tx_1 を追加することとなることを意味する。ブロックチェーンノード 104 はまた、トランザクション Tx_1 をネットワーク 106 内の 1 つまたは複数の他のブ

50

ロックチェーンノード104にフォワードして、トランザクション T_{x1} がネットワーク106全体に伝搬されるようにする。 T_{x1} が妥当性確認されてブロックチェーン150に含まれると、これは、 T_{x0} からのUTXO₀を使用済みとして定義する。 T_{x1} は、未使用トランザクション出力203を使用する場合にのみ有効になり得ることに留意されたい。別のトランザクション152によってすでに使用された出力を使用しようとする場合、 T_{x1} は、他のすべての条件が満たされたとしても無効になる。したがって、ブロックチェーンノード104はまた、先行するトランザクション T_{x0} 内の参照されたUTXOがすでに使用済みであるかどうか（すなわち、それが別の有効なトランザクションへの有効な入力をすでに形成したかどうか）をチェックする必要がある。これは、ブロックチェーン150がトランザクション152に定義された順序を課することが重要である1つの理由である。実際には、所与のブロックチェーンノード104は、どのトランザクション152内のどのUTXO₂₀₃が使用されたかをマーキングする別個のデータベースを維持し得るが、最終的には、UTXOが使用されたかどうかを定義するものは、ブロックチェーン150内の別の有効なトランザクションへの有効な入力をすでに形成しているかどうかである。

10

【0053】

所与のトランザクション152のすべての出力203において指定された総額が、そのすべての入力202によって指し示された総額よりも大きい場合、これは、ほとんどのトランザクションモデルにおいて無効性の別の根拠となる。したがって、そのようなトランザクションは、伝搬されることも、ブロック151に含まれることもないであろう。

20

【0054】

UTXOベースのトランザクションモデルでは、所与のUTXOが全体として使用される必要があることに留意されたい。UTXOにおいて使用済みとして定義された額の一部分は、別の一部が使用されていても、「後に残す」ことはできない。しかしながら、次のトランザクションの複数の出力間でUTXOからの額を分割することはできる。例えば、 T_{x0} 内のUTXO₀において定義された額を、 T_{x1} 内の複数のUTXO間で分割することができる。したがって、アリスが、UTXO₀において定義された額のすべてをボブに与えたくない場合、アリスは、リマインダを使用して、 T_{x1} の第2の出力において自分自身に残りを与えるか、または別の当事者に支払うことができる。

【0055】

実際には、アリスはまた、通常、アリスのトランザクション104をブロック151に成功裏に含めるビットコインノード104に対する手数料を含める必要がある。アリスがそのような手数料を含めない場合、 T_{x0} は、ブロックチェーンノード104によって拒否され得、したがって、技術的に有効であっても、伝搬されず、ブロックチェーン150に含まれない可能性がある（ノードプロトコルは、ブロックチェーンノード104が望まない場合にトランザクション152を受け入れることを強制しない）。いくつかのプロトコルでは、トランザクション手数料は、それ自体の別個の出力203を必要としない（すなわち、別個のUTXOを必要としない）。代わりに、所与のトランザクション152の入力（複数可）202によって指し示される総額と出力（複数可）203で指定されている総額との間の任意の差が、トランザクションを公開するブロックチェーンノード104に自動的に与えられる。例えば、UTXO₀へのポインタが T_{x1} への唯一の入力であり、 T_{x1} は唯一の出力UTXO₁を有するとする。UTXO₀において指定されたデジタル資産の額がUTXO₁において指定された額よりも大きい場合、その差分は、UTXO₁を含むブロックを生成するためのプルーフオブワーク競争に勝つノード104によって割り当てられ得る。しかしながら、代替的にまたは追加的に、トランザクション手数料がトランザクション152のUTXO₂₀₃のうちのそれ自体の1つにおいて明示的に指定され得ることは必ずしも除外されない。

30

40

【0056】

アリスおよびボブのデジタル資産は、ブロックチェーン150内のどこかで任意のトランザクション152においてそれらにロックされたUTXOから構成される。したがって

50

、典型的には、所与の当事者103の資産は、ブロックチェーン150全体にわたる様々なトランザクション152のUTXO全体に散在している。ブロックチェーン150内のどこにも、所与の当事者103の総残高を定義する数字は記憶されない。クライアントアプリケーション105におけるウォレット機能の役割は、それぞれの当事者にロックされ、別の前方のトランザクションでまだ使用されていない様々なUTXOのすべての値と一緒に照合することである。これは、ビットコインノード104のいずれかに記憶されたブロックチェーン150のコピーにクエリを行うことによって行うことができる。

【0057】

スクリプトコードは、多くの場合、概略的に（すなわち、正確な言語を使用せずに）表されることに留意されたい。例えば、特定の機能を表すためにオペレーションコード（オペコード）が使用され得る。「OP_...」は、スクリプト言語の特定のオペコードを指す。例として、OP_RETURNは、ロックスクリプトの最初にOP_FALSEが先行するときに、トランザクション内にデータを記憶することができ、それによってデータをブロックチェーン150内に不変的に記録することができる、トランザクションの使用不可能な出力を作成するスクリプト言語のオペコードである。例えば、データは、ブロックチェーンに記憶することが望まれる文書を含み得る。

【0058】

典型的には、トランザクションの入力は、公開鍵P_Aに対応するデジタル署名を含む。実施形態において、これは、楕円曲線secp256k1を使用するECDSAに基づく。デジタル署名は、データの特定の一部分に署名する。いくつかの実施形態では、所与のトランザクションについて、署名は、トランザクション入力の一部、およびトランザクション出力の一部または全部に署名する。署名された出力の特定の部分は、SIGHASHフラグに依存する。SIGHASHフラグは、通常、どの出力が署名されるかを選択するために署名の最後に含まれる4バイトコードである（したがって、署名時に固定される）。

【0059】

ロックスクリプトは、典型的には、それぞれのトランザクションがロックされる当事者の公開鍵を含むという事実を指して、「scriptPubKey」と呼ばれることがある。ロック解除スクリプトは、典型的には、それが対応する署名を供給するという事実を指して「scriptSig」と呼ばれることがある。しかしながら、より一般的には、UTXOが償還されるための条件が署名を認証することを含むことは、ブロックチェーン150のすべてのアプリケーションにおいて必須ではない。より一般的には、スクリプト言語を使用して、任意の1つまたは複数の条件を定義することができる。したがって、より一般的な用語「ロックスクリプト」および「ロック解除スクリプト」が好まれ得る。

【0060】

図1に示すように、アリスおよびボブのコンピュータ機器102a、120bの各々上のクライアントアプリケーションは、それぞれ、追加の通信機能を含み得る。この追加の機能により、（いずれかの当事者または第三者の扇動で）アリス103aは、ボブ103bと別個のサイドチャネル107を確立することができる。サイドチャネル107は、ブロックチェーンネットワークとは別でのデータの交換を可能にする。そのような通信は、「オフチェーン」通信と呼ばれることがある。例えば、これは、当事者の一方がトランザクションをネットワーク106にブロードキャストすることを選択するまで、トランザクションが（まだ）ブロックチェーンネットワーク106に登録されることなく、またはチェーン150上に進むことなく、アリスとボブとの間でトランザクション152を交換するために使用され得る。このようにトランザクションを共有することは、「トランザクションテンプレート」の共有と呼ばれることがある。トランザクションテンプレートは、完全なトランザクションを形成するために必要とされる1つまたは複数の入力および／または出力を欠いていてもよい。代替的にまたは追加的に、サイドチャネル107は、鍵、交渉された額または条件、データコンテンツなどの任意の他のトランザクション関連データを交換するために使用され得る。

10

20

30

40

50

【0061】

サイドチャネル107は、ブロックチェーンネットワーク106と同じパケット交換ネットワーク101を介して確立され得る。代替的にまたは追加的に、サイドチャネル301は、モバイルセルラーネットワークなどの異なるネットワーク、またはローカルワイヤレスネットワークなどのローカルエリアネットワーク、またはさらにはアリスのデバイス102aとボブのデバイス102bとの間の直接のワイヤードまたはワイヤレスリンクを介して確立され得る。一般に、本明細書のどこでも、参照されるサイドチャネル107は、「オフチェーン」すなわちブロックチェーンネットワーク106とは別でデータを交換するための1つまたは複数のネットワーキング技術または通信媒体を介した任意の1つまたは複数のリンクを含み得る。2つ以上のリンクが使用される場合、全体としてのオフチェーンリンクの束または集合は、サイドチャネル107と呼ばれ得る。したがって、アリスおよびボブがサイドチャネル107上で情報またはデータの特定の部分などを交換している場合、これは、これらのデータの部分のすべてが全く同じリンクまたは同じタイプのネットワーク上で送信されなければならないことを必ずしも意味するものではないことに留意されたい。

10

【0062】

クライアントソフトウェア

図3Aは、本開示の方式の実施形態を実装するためのクライアントアプリケーション105の例示的な実装を示す。クライアントアプリケーション105は、トランザクションエンジン401およびユーザインターフェース(UI)層402を含む。トランザクションエンジン401は、上述した方式にしたがって、および、さらに詳細に簡潔に説明されるように、例えば、トランザクション152を定式化し、サイドチャネル301上でトランザクションおよび／または他のデータを受信および／または送信し、および／またはトランザクションを1つまたは複数のノード104に送信してブロックチェーンネットワーク106全般に伝搬させるために、クライアント105の基本的なトランザクション関連機能を実装するように構成される。

20

【0063】

UI層402は、機器102のユーザ出力手段を介してそれぞれのユーザ103に情報を出力すること、および機器102のユーザ入力手段を介してそれぞれのユーザ103から入力を受信することを含む、それぞれのユーザのコンピュータ機器102のユーザ入力／出力(I/O)手段を介してユーザインターフェースをレンダリングするように構成される。例えば、ユーザ出力手段は、視覚出力を提供するための1つまたは複数のディスプレイスクリーン(タッチスクリーンまたは非タッチスクリーン)、オーディオ出力を提供するための1つまたは複数のスピーカ、および／または触覚出力を提供するための1つまたは複数の触覚出力デバイスなどを含み得る。ユーザ入力手段は、例えば、1つまたは複数のタッチスクリーン(出力手段のために使用されるものと同一または異なる)の入力アレイ、マウス、トラックパッド、またはトラックボールなどの1つまたは複数のカーソルベースのデバイス、発話または音声入力を受信するための1つまたは複数のマイクロホンおよび発話または音声認識アルゴリズム、手動または身体ジェスチャの形態で入力を受信するための1つまたは複数のジェスチャベースの入力デバイス、あるいは1つまたは複数の機械式ボタン、スイッチ、またはジョイスティックなどを含み得る。

30

40

【0064】

本明細書の様々な機能は、同じクライアントアプリケーション105に統合されるものとして説明され得るが、これは、必ずしも限定的ではなく、代わりに、それらは、2つ以上の別個のアプリケーション一式で実装され得、例えば、一方が他方へのプラグインであるか、またはAPI(アプリケーションプログラミングインターフェース)を介してインターフェースしていることに留意されたい。例えば、トランザクションエンジン401の機能が、UI層402とは別個のアプリケーションにおいて実装されてもよいし、トランザクションエンジン401などの所与のモジュールの機能が、2つ以上のアプリケーション間で分割されてもよい。説明された機能のうちのいくつかまたはすべてが、例えばオペ

50

レーティングシステム層で実装され得ることも除外されない。本明細書のどこでも単一のまたは所与のアプリケーション 105 などを参照する場合、これは単なる例であり、より一般的には、説明される機能は任意の形態のソフトウェアで実装され得ることが理解されよう。

【0065】

図 3 B は、アリスの機器 102 a 上のクライアントアプリケーション 105 a の UI 層 402 によってレンダリングされ得るユーザインターフェース (UI) 500 の例のモックアップを与える。同様の UI が、クライアント 105 b によってボブの機器 102 b 上に、または任意の他の当事者の機器上にレンダリングされ得ることが理解されよう。

【0066】

例示として、図 3 B は、アリスの視点からの UI 500 を示す。UI 500 は、ユーザ出力手段を介して別個の UI 要素としてレンダリングされる 1 つまたは複数の UI 要素 501、502、502 を含み得る。

【0067】

例えば、UI 要素は、異なるオンスクリーンボタン、またはメニュー内の異なるオプションなどであり得る 1 つまたは複数のユーザ選択可能要素 501 を含み得る。ユーザ入力手段は、ユーザ 103 (この場合、アリス 103 a) が、スクリーン上の UI 要素をクリックもしくはタッチすることまたは所望のオプションの名前を言うことなどによって、オプションのうちの 1 つを選択または他の方法で動作させることを可能にするように構成される (注意: 本明細書で使用される「手動」という用語は、自動との対比を意味し、必ずしも 1 つまたは複数の手の使用に限定されない)。オプションにより、ユーザ (アリス) は、トランザクション内に埋め込まれる署名を生成することができる。

【0068】

代替的にまたは追加的に、UI 要素は、1 つまたは複数のデータ入力フィールド 502 を含み得、それを通して、ユーザは、署名内に埋め込まれるデータを入力することができる。これらのデータ入力フィールドは、ユーザ出力手段、例えばオンスクリーンを介してレンダリングされ、データは、ユーザ入力手段、例えば、キーボードまたはタッチスクリーンを通してフィールドに入力され得る。代替的に、データは、例えば、音声認識に基づいて、口頭で受信され得る。

【0069】

代替的にまたは追加的に、UI 要素は、ユーザに情報を出力するために出力される 1 つまたは複数の情報要素 503 を含むことができる。例えば、これ／これらは、スクリーン上にまたは音声でレンダリングされ得る。

【0070】

様々な UI 要素をレンダリングし、オプションを選択し、データを入力する特定の手段は重要ではないことが理解されよう。これらの UI 要素の機能については、以下でより詳細に説明する。図 3 に示される UI 500 は、単に図式化されたモックアップであり、実際には、簡潔にするために図示されていない 1 つまたは複数のさらなる UI 要素を含み得ることも理解されよう。

【0071】

暗号プレリナリ (CRYPTOGRAPHY RELIMINARIES)

【0072】

E C D S A - 楕円曲線群

【数 1】

$(E, \oplus)$

は、有限体

【数 2】

$$\mathbb{F}_p$$

上の巡回楕円曲線群であり、 p は素数である。 E の要素の数は n であり、 n は素数である。 $G \in E$ は、楕円曲線群の生成点であり、以下を意味する：

【数 3】

$$\forall Y \in E \exists i \in \{1, \dots, n\} : Y = i \cdot G.$$

【0073】

群演算

【数 4】

「 $\oplus$ 」

10

は、標準楕円曲線点加算であり、 $i \cdot G$ は、 G に対する群演算の i 回の反復を表す：

【数 5】

$$i \cdot G = \underbrace{G \oplus G \oplus \dots \oplus G}_{i\text{回}}$$

【0074】

以下において、文脈上他の意味に解すべき場合を除き、整数に対するすべての演算はモジュロ n である。

【0075】

楕円曲線デジタル署名アルゴリズム

鍵生成は以下に行われる：

- 1) 署名用の秘密鍵 $j \in \{1, \dots, n-1\}$ を選択する
- 2) 公開鍵は、 $Y = j \cdot G$ であり、 G は、生成点である。

【0076】

署名アルゴリズムは、秘密鍵 j メッセージおよびエフェメラル鍵 k を取り、署名を生成する：

30

- 3) ランダムな $k \in \{1, \dots, n-1\}$ (エフェメラル鍵) を選択する
- 4) $R = (r_x, r_y) = k \cdot G - E$ C点を計算する
- 5) $r = r_x \bmod n$ を計算する
- 6) $r = 0$ である場合、ステップ3に進む
- 7) 署名 $s = k^{-1} (e + j r)$ を生成する。ここで、 $e = \text{hash}(m)$ である
- 8) $s = 0$ である場合、ステップ3に進む
- 9) $[r, s]$ をメッセージ m の署名として出力する

【0077】

次いで、検証アルゴリズムは、署名およびメッセージを取り、署名者の公開鍵を使用し、 r を再構築し、署名で与えられる r 値を検証する。

40

- 1) $e = \text{hash}(m)$ を計算する
- 2) $k_1 = e s^{-1} \bmod n$ および $k_2 = r s^{-1} \bmod n$ を計算する
- 3) $Q = (q_x, q_y) = k_1 \cdot G + k_2 \cdot Y$ を計算する
- 4)

【数 6】

$$q_x \equiv r \bmod n$$

である場合、署名は有効である
そうでなければ無効である。

50

【 0 0 7 8 】

署名には以下の表記が使用される：

【数 7】

$$Sig_Y = [r_Y, s_Y]$$

ここで、 $[r_Y, s_Y]$ は、公開鍵 Y を使用して検証されたときに、有効な署名である。

【 0 0 7 9 】

共同検証可能ランダム秘密分散（J V R S S）方式

N 人の参加者が、この方式の参加者のうちの少なくとも $(i + 1)$ によってのみ再生成することができる共同シークレットを作成したいと望むと仮定する。共有シークレットを作成するために、以下のステップが行われる：

1) 参加者は、各参加者の一意のラベル i に合意する。各参加者 i は、 $(t + 1)$ 個の乱数

【数 8】

$$a_{ij} \in_R \mathbb{Z}_n \setminus \{0\}, \forall j = 0, \dots, t$$

を生成し、ここで、 $\in_R$ は、集合

【数 9】

$$\mathbb{Z}_n \setminus \{0\}$$

のランダムに生成された要素を意味し、

【数 1 0】

$$\mathbb{Z}_n \setminus \{0\}$$

は、集合 $\{1, \dots, n - 1\}$ の表記である。次いで、各参加者は、 $i = 1, \dots, N$ に対して、次数 t のシークレット多項式 (secret polynomial)

【数 1 1】

$$f_i(x) = a_{i0} + a_{i1}x + \dots + a_{it}x^t \bmod n$$

を有する。なお、これ以降 $\bmod n$ の表記は省略され、整数に対するすべての算術演算はモジュロ n で行われると仮定する。

2) 各参加者 i は、例えば、参加者 j とのセキュアな通信チャンネルだけを使用して、値 $f_i(j)$ を参加者 j に送信する。

3) 各参加者 i は、共有シークレット多項式の自身のプライベートなシークレットシェアを以下のように計算する：

【数 1 2】

$$a_i := \sum_{j=1}^N f_j(i)$$

【 0 0 8 0 】

共有シークレットシェアは、形式 (i, a_i) を有する点であり、ここで、 i は、方式における参加者ラベルである。 a のシークレットシェアを生成するためのこの方法は、ステップ 1 ~ 3 に説明されるように、本明細書では、参加者 i について、 $a_i = J V R S S(i)$ と表記される。「J V R S S」は、典型的に、「Joint verification random

10

20

30

40

50

secret sharing」の略であり、ステップ 4 および 5 も含むことに留意されたい。しかしながら、本明細書を通して、JVRS S は、少なくともステップ 1 ～ 3 を実行することを意味するものと解釈され、ステップ 4 および 5 は、オプションのステップである。

【0081】

参加者が、共有多項式を生成しているので、参加者はそれぞれ、他の参加者がすべての参加者に正しい情報を共有したこと、およびすべての参加者が同じ共有多項式を有することを検証することができる。これは次のように行われる。

4) 各参加者 i は、 $k = 0, \dots, t$ に対して、難読化された係数 $a_{ik} \cdot G$ をすべての参加者にブロードキャストする

5) 各参加者 i は、 $f_j(i) \cdot G$ を計算し、以下を検証することによって、各参加者 j が多項式点 $f_j(i)$ を正しく計算したことをチェックする：

10

【数 1 3】

$$f_j(i) \cdot G \stackrel{?}{=} \sum_{k=0}^t i^k (a_{jk} \cdot G) \quad \forall j = 1, \dots, N$$

【0082】

この式が各多項式に対して成立することをすべての参加者が確認すると、グループは、それらがすべて同じ共有多項式を作成したことを集合的に確信することができる。

【0083】

20

共有シークレットの再構成

参加者が、共有多項式のゼロ次である共有シークレット a を再構築したいと望むと仮定する。

【数 1 4】

$$(1, a_1), \dots, ((t+1), a_{t+1})$$

の形態のこの多項式上の $(t+1)$ 個の点を与えられると、共有シークレット a を見つけるために、以下が計算され、

【数 1 5】

30

$$\text{interpolate}(a_1, \dots, a_{t+1}) = \left(\sum_{l=1}^{t+1} a_l \prod_{\substack{1 \leq j \leq (t+1), \\ j \neq l}} (-j) (l-j)^{-1} \right) = a$$

これは、「ラグランジュ補間」として知られる一般式から導出される。

【0084】

公開鍵計算

JVRS S のステップ 4 で共有された $j = 1, \dots, N$ に対して、 N 個のゼロ次のプライベート多項式係数の公開鍵 $a_{j0} \cdot G$ が与えられると、各参加者は、共有シークレット a に対応する

40

【数 1 6】

$$P = a \cdot G = \sum_{j=1}^N a_{j0} \cdot G$$

を使用して、共有公開鍵 P を計算する。

【0085】

50

共有シークレットの加算

例えば、いずれのエンティティも個々のシークレットを知ることなく、N人の参加者のグループ間で共有される2つの共有シークレットの加算を計算するために、各シークレット多項式が次数 t を有する場合、以下のステップが行われる：

- 1) 第1の共有シークレット a を生成する。ここで、参加者 i のシェアは、しきい値が $(t+1)$ の $i = 1, \dots, N$ に対して、 $a_i = J V R S S(i)$ で与えられる。
 - 2) 第2の共有シークレット b を生成する。ここで、参加者 i のシェアは、 $b_i = J V R S S(i)$ で与えられ、しきい値は $(t+1)$ である。
 - 3) 各参加者 i は、自身の加法的シェア (additive share) を計算する：
- 【数 1 7】

$$v_i = a_i + b_i \bmod n$$

10

4) すべての参加者は、自身の加法的シェア v_i を他のすべての参加者にブロードキャストする。

5) 各参加者は、シェア v_i のうちの少なくとも $(t+1)$ 個にわたって補間して、以下を計算する：

【数 1 8】

$$v = \text{interpolate}(v_1, \dots, v_{t+1}) = a + b$$

20

【0 0 8 6】

共有シークレットの加算のためのこの方法は、参加者 i について、 $A D D S S(i)$ と表記され、この結果、各参加者 i は、 $v = (a + b)$ を知ることとなる。

【0 0 8 7】

共有シークレットの積

N人の参加者のグループ間で共有される2つの共有シークレットの積を計算するために、各シークレット多項式が次数 t を有する場合、グループは以下のステップを行う：

- 1) 第1の共有シークレット a を生成する。ここで、参加者 i のシェアは、 $i = 1, \dots, N$ に対して、 $a_i = J V R S S(i)$ で与えられる。共有シークレット多項式は次数 t を有し、これは、それを再作成するために $(t+1)$ 人の参加者が必要とされることを意味する。
- 2) 第2の共有シークレット b を生成する。ここで、参加者 i のシェアは、 $b_i = J V R S S(i)$ で与えられ、共有シークレット多項式は、この場合も次数 t を有する。
- 3) 各参加者は、以下を使用して、自身の乗法的シェア (multiplicative share) μ_i を計算する：

30

【数 1 9】

$$\mu_i = a_i b_i$$

40

4) すべての参加者は、自身の乗法的シェア μ_i をすべての他の参加者にブロードキャストする。

5) 各参加者は、0においてシェア μ_i のうちの少なくとも $(2t+1)$ 個にわたって補間して、以下を計算する：

【数 2 0】

$$\mu = \text{interpolate}(\mu_1, \dots, \mu_{2t+1}) = ab$$

【0 0 8 8】

2つの共有シークレットの積を計算するためのこの方法は、本明細書では、参加者 i に

50

ついて、 $\mu = a \cdot b = \text{PROSS}(i)$ と表記される。

【0089】

共有シークレットの逆数 (inverse)

共有シークレット a の逆数を計算するために、以下のステップが行われる：

1) すべての参加者は、共有シークレットの積 $\text{PROSS}(i)$ を計算し、その結果は、 $\mu = a \cdot b \bmod n$ である。

2) 各参加者は、 μ のモジュラ逆数を計算し、その結果、以下となる：

【数21】

$$\mu^{-1} = (ab)^{-1} \bmod n$$

10

3) 各参加者 i は、以下を計算することによって、自身の逆数シークレットシェア (inverse secret share) を計算する。

【数22】

$$a_i^{-1} = \mu^{-1} b_i$$

【0090】

共有シークレットの逆数を計算するためのこの方法は、参加者 i について、以下のよう

20

に表記される：

【数23】

$$a_i^{-1} = \text{INVSS}(i)$$

【0091】

共有秘密鍵の生成および検証

$N \geq 2t + 1$ 人の参加者の間の共有秘密鍵 a を計算するために、そのうちの $t + 1$ 人が署名を作成することが求められる場合、参加者は、上述したように公開鍵計算およびしきい値 $t + 1$ を用いて JVRS を実行する。その結果、すべての参加者 $i = 1, \dots, N$ が秘密鍵シェア a_i と、対応する共有公開鍵 $P = (a \cdot G)$ とを有する。

30

【0092】

エフェメラル鍵シェアの生成

署名において必要とされるような、エフェメラル鍵シェアおよび対応する r を生成するために、しきい値 $(t + 1)$ の共有秘密鍵 a を有するサイズ N のグループは、以下のステップを実行する：

1) 共有シークレットの逆数シェア (inverse share)

【数24】

$$k_i^{-1} = \text{INVSS}(i)$$

40

を生成する。ここで、それを再作成するためには $(t + 1)$ 個のシェアが必要とされる。

2) 各参加者は、以下を計算する。

【数25】

$$(x, y) = \sum_{i=1}^N (k_{i0} \cdot G)$$

3) k_i の検証で共有された難読化された係数を使用して、以下を計算する。

【数26】

50

$$r = x \bmod n$$

4) 各参加者 i は、以下を記憶する。

【数 2 7】

$$(r, k_i^{-1})$$

【0093】

非最適署名生成

10

少なくとも $2t + 1$ 人の参加者がメッセージに対する署名を作成したいと思っており、参加者のうちの 1 人がこれを調整することを選択すると仮定する。共有秘密鍵を有するグループによる署名を作成するために、以下のステップが行われる。

1) コーディネータは、少なくとも 1 つの $2t + 1$ 人の参加者に、メッセージに対する署名を要求する。

2) 各参加者 i は、前のセクションで計算されたエフェメラル鍵

【数 2 8】

$$(r, k_i^{-1})$$

20

を復元する。すべてのユーザは、同じエフェメラル鍵に対応するシェアを使用しなければならない。

3) 各参加者は、メッセージダイジェスト $e = \text{SHA-256}(\text{SHA-256}(\text{message}))$ を計算する。

4) 各参加者 i は、自身の署名シェア s_i を計算する：

【数 2 9】

$$s_i = k_i^{-1}(e + a_i r) \bmod n$$

ここで、 a_i は、各参加者の秘密鍵シェアである。

30

5) 各参加者は、自身の署名シェア (r, s_i) をコーディネータに送信する。

6) コーディネータは、 $2t + 1$ 個の署名シェアを受信すると、

【数 3 0】

$$s = \text{interpolate}(s_1, \dots, s_{2t+1})$$

を計算し、この署名を (r, s) として出力する。

7) コーディネータは、標準 ECDSA 検証を使用して署名を検証する。これが失敗した場合、シェアのうちの少なくとも 1 つが誤っているはずであり、署名生成アルゴリズムを再度実行する必要がある。

40

【0094】

ディフィーヘルマン (DH) 鍵交換

2 つの当事者は、以下の方法で対称シークレット鍵を作成することによって、セキュアな通信チャネルを確立し得る。アリスとボブが共有シークレット鍵 (shared secret key) を作成したいと望み、アリスが、公開鍵 $PK_A = sk_A \cdot G$ に対応する秘密鍵 sk_A を知っており、ボブが、ボブの公開鍵 $PK_B = sk_B \cdot G$ に対応する秘密鍵 sk_B を知っているとして仮定する。

【0095】

共有シークレット鍵を見つけるために、以下のステップを行う。

1. アリスは、ディフィーヘルマン鍵 $sk_{AB} = sk_A \cdot PK_B$ を計算する。

50

2. ボブは、ディフィーヘルマン鍵 $s_{k_{AB}} = s_{k_B} \cdot P_{K_A}$ を計算する。

【0096】

共有シークレット鍵を確立するための別の方法は、国際公開第2017/145016号に記載されており、この方法では、事前合意されたメッセージがDH鍵に追加され、新しい鍵が作成される。このメッセージは、送信される新しい通信ごとに変更することができ、決定論的鍵のセットを作成する。例えば、メッセージは、 $m = \text{hash}(\text{date} \parallel \text{time})$ であり得る。次いで、アリスは、メッセージを使用して秘密鍵 $s_{k_1} = s_{k_A} + \text{hash}(\text{date} \parallel \text{time})$ を生成し、同様に、ボブは、秘密鍵 $s_{k_1} = s_{k_B} + \text{hash}(\text{date} \parallel \text{time})$ を生成することができる。そのため、アリスとボブの両方が、共有秘密鍵 $s_{k_{AB1}} = s_{k_{A1}} \cdot P_{K_{B1}} = s_{k_{B1}} \cdot P_{K_{A1}}$ を生成することができる。

10

【0097】

HDウォレット

階層的決定論的ウォレットは、ビットコイン改善提案32 (BIP32: Bitcoin Improvement Proposal 32) ウォレットがその特定のタイプであり、多くの鍵が単一の入力から導出され得る決定論的ウォレットである。入力、シードと呼ばれる何らかのランダムエントロピーであり、そこからマスタ鍵が導出される。次いで、マスタ鍵は、図2に示されるように、複数の子鍵を導出するために使用される。

【0098】

BIP32では、マスタ秘密鍵は、シードのHMAC-SHA512の結果の左の32

20

【数31】

$$sk_{master} = \text{HMAC-SHA512}_L('Bitcoin Seed', seed)$$

であり、チェーンコードは、

【数32】

$$c_{master} = \text{HMAC-SHA512}_R('Bitcoin Seed', seed)$$

30

であり、すべての子鍵はこれらから導出することができ、ここで、

【数33】

$$\text{HMAC-SHA512}(c, K) = \text{SHA512}(c \oplus \text{opad} \parallel \text{SHA512}((c \oplus \text{ipad}) \parallel K))$$

は、SHA512ハッシュ関数を使用するHMACである。上記の式において、opadは、ブロックサイズの外側パディングであり、ipadは、ブロックサイズの内部パディングである。

【0099】

HMACは、2つの入力、すなわち、cおよびKを必要とする。簡潔にするために、かつユーザが単一のシードを覚えるまたは記憶することのみを要求されるように、BIP32プロトコルは、第1の入力を文字列「Bitcoin Seed」として設定、すなわち、c = 'Bitcoin Seed' とする。これは、HDウォレットを生成するための1つの例示的プロトコルであり、異なるプロトコルは、異なる入力、例えば、2つのランダムに生成されたシードを必要とし得ることが理解されよう。言い換えると、文字列「Bitcoin Seed」の使用は、HDウォレットを生成するための必要要件ではない。

40

【0100】

親秘密鍵 $s_{k_{parent}}$ から、強化された子秘密鍵 $s_{k_{child}}$ を計算するための式は、以下の通りである：

【数34】

50

$$sk_{child} = sk_{parent} + HMAC-SHA512_L(c_{parent}, sk_{parent} || index)$$

ここで、 c_{parent} は、親チェーンコードであり、 $0 \leq index < 2^{31}$ は子インデックスであり、 $HMAC-SHA512_L$ は、 $SHA-512$ ハッシュ関数を用いて計算された $HMAC$ 関数の結果の左の32バイトである。子公開鍵のための対応する式は、この式にベースポイント G を単純に点乗算 (point multiply) することによって導出される。子チェーンコード c_{child} は、 $HMAC$ 関数の結果の右の32バイトであると定義される：

【数35】

$$c_{child} = HMAC-SHA512_R(c_{parent}, sk_{parent} || index)$$

10

【0101】

親公開鍵 pk_{parent} および親秘密鍵 sk_{parent} から、強化されていない子秘密鍵 sk_{child} を計算するための式は以下である：

【数36】

$$sk_{child} = sk_{parent} + HMAC-SHA512_L(c_{parent}, pk_{parent} || index)$$

20

ここで、 c_{parent} は、親チェーンコードであり、 $2^{31} \leq index < 2^{32}$ は子インデックスであり、 $HMAC-SHA512$ は、 $SHA-512$ ハッシュ関数を用いて計算された $HMAC$ 関数である。強化された鍵と同様に、強化されていない鍵の子チェーンコード c_{child} は、 $HMAC$ 関数の結果の右の32バイトであると定義される：

【数37】

$$c_{child} = HMAC-SHA512_R(c_{parent}, pk_{parent} || index)$$

【0102】

この第2のタイプの子鍵により、親公開鍵およびチェーンコードの知識を有する者であれば、以下の式を使用して、子公開鍵を導出することができる：

30

【数38】

$$pk_{child} = pk_{parent} + HMAC-SHA512_L(c_{parent}, pk_{parent} || index) \cdot G$$

【0103】

これを使用して、外部当事者は、必要に応じて様々な支払いアドレスを導出し、通信および記憶の回数を減らすと同時に、鍵の再使用を回避することができる。

【0104】

一般に、HDウォレットは、秘密鍵-公開鍵ペアの何らかの階層ツリー状構造を生成する必要がある。これにより、1つのシードからすべて再生成することができる多数の鍵ペアが提供される。

40

【0105】

しきい値デジタル署名

図4は、本発明のいくつかの実施形態による、デジタル署名を生成するための例示的なシステム400を示す。システムは、各々が共有秘密鍵のそれぞれのシェアを有する複数の参加者401を含む。共有秘密鍵は、秘密分散方式 (secret sharing scheme)、例えば、JRSSSまたはShamirの秘密分散方式を使用して生成され得る。共有秘密鍵は、ディーラーを伴う方式を使用して、またはディーラーを伴わない方式を使用して生成されていてもよい。図4には3人の参加者のみが示されているが、一般に、システ

50

ムは任意の数の参加者 401 を含み得る。システムはまた、検証当事者 402、すなわち署名検証当事者と、コーディネータ 404 とを含む。コーディネータ 404 は、しきい値数の署名シェアに基づいて署名を生成するように構成され、各署名シェアは、それぞれの参加者によって生成される。コーディネータ 404 は、図 4 では別個のものとして示されているが、いくつかの例では、コーディネータは、参加者のうちの 1 つ、例えば第 1 の参加者 401 a と同じエンティティであってもよい。いくつかの例では、システムは、1 つまたは複数のブロックチェーンノード 104 を含む。

【0106】

各参加者 401、検証当事者 402、およびコーディネータ 404 は、それぞれのコンピューティング機器（図示せず）を操作する。各それぞれのコンピューティング機器は、1 つまたは複数のプロセッサ、例えば 1 つまたは複数の中央処理装置（CPU）、アクセラレータプロセッサ（GPU）、特定用途向けプロセッサ、および／またはフィールドプログラマブルゲートアレイ（FPGA）を含むそれぞれの処理装置を備える。それぞれのコンピューティング機器はまた、メモリ、すなわち、1 つまたは複数の非一時的コンピュータ可読媒体の形態のコンピュータ可読ストレージを備え得る。メモリは、1 つまたは複数のメモリ媒体、例えば、ハードディスクなどの磁気媒体、ソリッドステートドライブ（SSD）、フラッシュメモリもしくは EEPROM などの電子媒体、および／または光ディスクドライブなどの光学媒体を採用する 1 つまたは複数のメモリユニットを備え得る。それぞれのコンピューティング機器は、少なくとも 1 つのユーザ端末、例えば、デスクトップもしくはラップトップコンピュータ、タブレット、スマートフォン、またはスマートウォッチなどのウェアラブルデバイスを含み得る。代替的にまたは追加的に、それぞれのコンピューティング機器は、ユーザ端末を介してアクセスされるクラウドコンピューティングリソースなどの 1 つまたは複数の他のネットワーク化されたリソースを含み得る（クラウドコンピューティングリソースは、1 つまたは複数のサイトで実装される 1 つまたは複数の物理サーバデバイスのリソースを含む）。システム 400 の当事者によって実行されるものとして説明される任意の行為は、その当事者によって操作されるそれぞれのコンピューティング装置によって実行され得ることが理解されよう。

【0107】

本発明は、ブロックチェーンの文脈での使用のみに限定されないが、以下では、第 1 の参加者 401 a を、図 1～図 3 を参照して説明されたアリス 103 a と同等として説明する。すなわち、いくつかの例では、アリス 103 a は第 1 の参加者 401 a である。検証当事者 402 は、以下ではキャロル 402 と呼ばれる。

【0108】

これらの実施形態では、アリス 401 a は、メッセージのための署名のシェアを生成し、自身がその署名シェアを生成したことをキャロル 402 に証明したいと思う。

【0109】

アリス 401 a は、署名されるべきメッセージ、例えば、ブロックチェーントランザクション、文書、またはコントラクトなどのうちのいくつかまたは一部を取得する。アリス 401 a は、自らメッセージを生成してもよいし、アリス 401 a は、例えば、別の参加者 401 またはコーディネータ 404 からメッセージを受信してもよい。アリス 401 a は、外部データ項目も取得する。アリス 401 a は、例えば、名前、パスポート番号、公開鍵などの外部データ項目をすでに有していてもよいし、アリス 401 a は、外部データ項目を生成してもよい。例えば、以下でより詳細に説明するように、外部データ項目は、アリス 401 a によって生成された署名（「第 2 の署名」）であり得る。

【0110】

アリス 401 a は、外部データ項目のハッシュに基づいてエフェメラル秘密鍵シェアを生成する。各他の参加者、例えば第 2 の参加者 401 b および第 3 の参加者 401 c も、それぞれのエフェメラル秘密鍵シェアを生成する。他の参加者は、それぞれの外部データ項目のハッシュに基づいてそれぞれのエフェメラル秘密鍵シェアを生成することが好ましいが、これは必須ではない。

10

20

30

40

50

【0111】

他の参加者のエフェメラル秘密鍵シェアがそれぞれの外部データ項目に基づくかどうかにかかわらず、各参加者のエフェメラル秘密鍵シェアは、他の参加者によって生成された入力（「データ項目」）に基づいて生成される。すなわち、アリスのエフェメラル秘密鍵シェアは、すべての他の参加者からのそれぞれのデータ項目に基づいて生成され、他の参加者のそれぞれのエフェメラル秘密鍵シェアは、アリスからのデータ項目の関数である。特定の例として、各参加者は、それぞれの外部データ項目に基づくそれぞれのデータ項目（例えば、多項式のゼロ次係数）を生成し得る。プライバシーを保護するために、データ項目は、秘密分散方式を使用して共有され得る。参加者は、参加者が他の参加者のインデックスで評価された多項式を共有する上述した J V R S S 方式を使用し得る。J V R S S 方式は、ランダムに生成されるのとは対照的に、外部データ項目に基づいて多項式のゼロ次係数を計算することによってわずかに修正される。これについては以下でより詳細に説明する。

10

【0112】

アリスのエフェメラル秘密鍵シェアは、ここではアリスの外部データ項目をハッシュした結果の関数である。外部データ項目のハッシュを生成するために使用されるハッシュ関数は、任意の適切なハッシュ関数、例えば、SHA 256、SHA 512 であり得、1つまたは複数のハッシュ関数を複数回適用することを含み得る。例えば、ハッシュ関数は、ダブルハッシュ関数、例えば、 $SHA256d(x) = SHA256(SHA256(x))$ であり得る。

20

【0113】

アリス 401a によって生成された署名シェアは、エフェメラル秘密鍵シェア、または特定の署名アルゴリズムによっては、エフェメラル秘密鍵シェアの逆数を必要とする。それぞれのエフェメラル秘密鍵シェアがそれぞれの参加者に対してプライベートなままであるために、例えば、アリス 103a のみがアリスのそれぞれのエフェメラル秘密鍵シェアを知るように、参加者は、秘密分散方式を使用してそれぞれのエフェメラル秘密鍵シェアを生成し得、すなわち、各参加者は、各々が同じシークレットのシェアを生成するのに十分な情報を他の参加者と共有はするが、シークレット自体を実際に生成することはない。この場合、共有シークレットはエフェメラル秘密鍵である。例えば、参加者は、J V R S S の修正バージョンを使用し得る。J V R S S のステップは上に詳述されている。この修正バージョン（J V R S S - A と呼ぶ）では、ステップ 1）において、参加者は、それぞれのプライベート多項式（private polynomial）のゼロ次係数 a_{i0} を、それぞれのデータ項目のハッシュ（例えば、ダブルハッシュ）と等しいものとして設定する。他の多項式係数は、通常の J V R S S の場合のようにランダムに選択されてもよい。次いで、J V R S S の残りのステップ 1）～ 5）を通常通りに行う。次いで、各参加者は、それぞれのエフェメラル秘密鍵シェアと、難読化されたデータ項目（すなわち、データ項目に対応する公開鍵）のセットとを、他の参加者ごとに 1 つずつ有する。J V R S S、またはむしろ J V R S S - A は、秘密分散方式の特定の例であり、他の方式、例えば、Shamir の秘密分散方式が使用されてもよいことに留意されたい。

30

【0114】

アリス 401a は、自身のデータ項目に対応する公開鍵を生成する。例えば、これは、「データ項目を生成点で難読化することによって行われる。アリス 401a は、各他の参加者 401 からそれぞれの公開鍵を取得し、これらの公開鍵を使用して、エフェメラル秘密鍵に対応するエフェメラル公開鍵を生成する。例えば、参加者は、J V R S S - A を使用して、エフェメラル公開鍵を生成し得る。「エフェメラル公開鍵」は、ここではエフェメラル公開鍵の x 構成要素（または x 座標）の省略表現として使用されていることに留意されたい。

40

【0115】

エフェメラル秘密鍵シェアおよびエフェメラル公開鍵（すなわち、共有エフェメラル秘密鍵に対応する公開鍵）を生成した後、アリス 401a は、自身の署名シェアを生成する

50

ことができる。アリスの署名シェアは、メッセージ（例えば、ブロックチェーントランザクション）と、共有秘密鍵のアリスのそれぞれのシェアと、（逆数）エフェメラル秘密鍵シェアと、エフェメラル公開鍵とに基づいて生成される。この場合も同様に、使用されている特定の署名方式に応じて、署名シェアは、エフェメラル秘密鍵の逆数に基づいて生成され得ることに留意されたい。いずれにしても、エフェメラル秘密鍵シェアが外部データ項目に基づいて生成されると仮定すると、その外部データ項目は、ここで、署名シェア内に埋め込まれている。

【0116】

次いで、アリス401aは、自身の署名シェアをコーディネータ404に送信し得、コーディネータは、アリスの署名シェアと、それぞれの参加者401によって生成された1つまたは複数のそれぞれの署名シェアとに基づいて第1の署名を生成することができる。コーディネータ404は、有効な署名を生成するためにしきい値数の署名シェアを必要とする。上記で簡単に述べたように、アリス401aは、実際にはコーディネータ404であってもよい。その場合、アリスは、自身の署名シェアをすでに入手しており、他の参加者401からそれぞれの署名シェアを受信する。

10

【0117】

したがって、署名は、この時点でアリス401aのみに知られている外部データ項目、例えば個人識別子を組み込む。外部データ項目自体が秘密（secret）である必要はないことに留意されたい。外部データ項目が署名内に埋め込まれるという事実は、最初は秘密に保たれることが好ましいが、これは必須ではない。例えば、外部データ項目は、それ自体1つまたは複数の当事者に知られているが、外部データ項目としてのその使用は知られていない、アリスの証明された公開鍵であってもよい。

20

【0118】

アリス401aは、第1の署名のシェアを生成したことを証明するために、署名をキャロル402が利用できるようにし得る。例えば、アリス401aは、署名をキャロル402に送信してもよいし、アリスは、署名を公開するか、または他の方法でブロードキャストしてもよい。他の例では、キャロル402は、異なる当事者、例えばコーディネータ404から署名を取得し得る。アリス401aはまた、メッセージをキャロル402に送信し得る。署名およびメッセージは、一緒に送信または公開されることが好ましい。

【0119】

アリス401aはまた、署名と同時に、または異なる時間に、例えば後の時間に、外部データ項目をキャロル402が利用できるようにし得る。署名と同じ方法で、または異なる方法で外部データ項目を利用できるようにし得る。例えば、アリス401aは、セキュアな通信チャネルを介して外部データ項目をキャロル402に送信し得る。アリス401aが署名シェアを生成したことを検証するために、キャロル402は、他の参加者401によって生成されたそれぞれのデータ項目に対応するそれぞれの公開鍵（例えば、ゼロ次係数を難読化する公開鍵）を必要とする。アリス401は、これらをキャロル402に送信し得るか、または他の参加者は、それぞれのエフェメラル公開鍵シェアをキャロル402に送信し得る。

30

【0120】

上述したように、外部データ項目は、別の署名であり得るか、または少なくとも別の署名を含み得る。その場合、アリス401aは、第2のメッセージを取得し、少なくとも第2のメッセージと「主秘密鍵」とに基づいて第2の署名を生成する。最も広い例では、「主（main）」は、単にラベルとして使用される。すなわち、主秘密鍵は、アリス401aによって所有される任意の秘密鍵であり得る。

40

【0121】

アリス401は、第2のメッセージの少なくとも一部を自ら生成し得る。追加的にまたは代替的に、アリス401aは、別の当事者、例えばキャロル402から、第2のメッセージの少なくとも一部を受信するか、または他の方法で取得し得る。すなわち、キャロル402が、第2のメッセージの一部または全部をアリス401aに送信してもよいし、ア

50

リス 4 0 1 a および キャロル 4 0 2 が、第 2 のメッセージの少なくとも一部に関して事前に合意していてもよい。例えば、リス 4 0 1 a および キャロル 4 0 2 は、第 2 の署名が生成された時間および／またはデータの指示を含めることに合意していてもよい。いくつかの例では、第 2 のメッセージは、第 1 のメッセージを含むか、またはそれに基づいて生成され得る。例えば、第 2 のメッセージは、第 1 のメッセージの始まりまたは終わりに連結された追加データを有する第 1 のメッセージを含み得る。

【 0 1 2 2 】

いくつかの例では、各参加者 4 0 1 は、同じ第 2 のメッセージを使用して、それぞれの第 2 の署名を生成する。すなわち、各参加者 4 0 1 は、第 2 のメッセージと、受け入れ可能な主秘密鍵とに基づいてそれぞれの第 2 の署名を生成する。

【 0 1 2 3 】

リス 4 0 1 a は、少なくとも第 2 の署名をキャロル 4 0 2 に送信し得るか、またはリス 4 0 1 a は、第 2 の署名を公開し得る。キャロル 4 0 2 が第 2 のメッセージをまだ入手していない場合、リス 4 0 1 a は、第 2 のメッセージをキャロル 4 0 2 に送信するか、または第 2 のメッセージを公開し得る。リス 4 0 1 a はまた、主秘密鍵に対応する主公開鍵をキャロル 4 0 2 に送信し得るか、または少なくとも、キャロル 4 0 2 が主公開鍵を取得し得る場所、例えば、証明機関によって公開され、主公開鍵がリス 4 0 1 a にリンクされていることを証明する証明書を記憶している位置を示し得る。

【 0 1 2 4 】

エフェメラル秘密鍵シェアは、少なくとも外部データ項目のハッシュ、例えば第 2 の署名のダブルハッシュに基づく（すなわち、その関数である）。エフェメラル秘密鍵シェアはまた、ランダムに生成されたソルト値、すなわち、外部データ項目のハッシュに追加された値に基づき得る。より具体的には、リスのエフェメラル秘密鍵シェアを生成するために使用されるデータ項目は、ソルト値に基づき得る。好ましくは、ソルト値は一度だけ使用され、すなわち、第 1 の署名の異なるインスタンスを生成するためには異なるソルト値が使用される。これらの例では、リス 4 0 1 a は、第 3 のメッセージとソルト値とに基づいて第 3 の署名を生成し得る。すなわち、ソルト値は、第 3 の署名を生成するための秘密鍵として使用される。第 3 のメッセージは、第 1 のメッセージおよび／または第 2 のメッセージに基づいて生成され得る。第 3 のメッセージは、第 2 のメッセージと同じであってもよい。リス 4 0 1 a は、第 3 の署名をキャロル 4 0 2 に送信し得る。リス 4 0 1 a が第 3 のメッセージを生成した場合、リスはそれもキャロル 4 0 2 に送信し得る。あるいは、キャロル 4 0 2 は、第 3 のメッセージをリス 4 0 1 a に送信している場合があり、その場合、リス 4 0 1 a は、それをキャロル 4 0 2 に送り返す必要はないが、送り返すことを選択してもよい。

【 0 1 2 5 】

第 3 の署名を生成する代わりに、リス 1 0 3 a は、ゼロ知識証明（zero-knowledge proof: ZKP）を使用してランダムソルト値の知識を証明し得る。当業者は、ZKP 自体に精通しているので、本明細書では詳細に説明しない。例示的な ZKP を以下に提供する。

【 0 1 2 6 】

オプションで、リス 4 0 1 a は、データ項目に対応するそれぞれの公開鍵、すなわち、リス 4 0 1 a によって生成された公開鍵および他の参加者から取得された公開鍵を使用して、ハッシュツリー、例えばマークルツリーを生成し得る。これらの公開鍵の各々は、ハッシュツリーのそれぞれのリーフハッシュを形成するためにハッシュされる。リス 4 0 1 a は、結果として得られたハッシュルート（例えば、マークルルート）をキャロル 4 0 2 に送信し得るか、またはハッシュルートを公開し得る。同時にまたは後の時間に、リス 4 0 1 a は、リスの公開鍵がハッシュツリーの要素であることを証明するために、ハッシュプルーフ（例えば、マークルプルーフ）をキャロル 4 0 2 に送信し得る。いくつかの例では、第 1 のメッセージは、ハッシュルートを含み、したがって、署名シェアを生成する各参加者は、同じハッシュルートを証明する（attest to）。すなわち、それ

10

20

30

40

50

らは、ハッシュルートを生成するために使用された公開鍵を証明する。例示的なハッシュルートを図 7 に示す。

【0127】

上述したように、本発明の実施形態は、ブロックチェーン150との併用に限定されない。しかしながら、そのようなものでは、第1のメッセージは、ブロックチェーントランザクションであり得る。例えば、アリス401aは、ブロックチェーントランザクションの一部または全部、例えばトランザクションの1つまたは複数の入力および／または1つまたは複数の出力に署名するために使用される署名のシェアを生成し得る。次いで、コーディネータ404は、アリスが署名していないトランザクションの入力に第1の署名を含め得る。トランザクションは、異なる当事者、例えばボブ103bおよび／またはキャロル402にロックされた出力を含み得、例えば、出力は、ボブ103bによって所有される公開鍵にロックされたpay-to-public-key (P2PK) またはpay-to-public-key-hash (P2PKH) 出力であり得る。第2のメッセージは、トランザクションを含み得る。第2のメッセージは、ブロックチェーン150に関するデータ、例えば、トランザクションが生成されたときのブロックチェーンの現在のブロック高さを含み得る。これらの例では、アリス401aまたはコーディネータ404は、ブロックチェーン150へのトランザクションを送信することによって、第1のメッセージをキャロル402が利用できるようにし得、そこから、キャロル402がアクセスし得る。これは図4に示されている。ハッシュルートは、ブロックチェーントランザクションの出力に含まれ得る。

10

20

【0128】

図5は、本発明のいくつかの実施形態による、署名シェアを生成するためにアリス401aによって行われ得るステップの例示的なシーケンスを示す。ステップのいくつか異なる順序で実行され得ることは理解されよう。ステップS501において、アリス401aは、第1のメッセージ、例えば、ブロックチェーントランザクションを取得する。ステップS502において、アリス401aは、外部データ項目、例えば、第2の署名を取得する。ステップS503において、アリス401aは、外部データ項目に基づいて、例えば、第2の署名のハッシュに基づいて、エフェメラル秘密鍵シェアを生成する。ステップS504において、アリス401aは、共有エフェメラル秘密鍵に対応するエフェメラル公開鍵を生成する。ステップS505において、アリス401aは、署名シェアを生成し、ステップS506において、アリスは、それをコーディネータ404に送信する。コーディネータ404が署名を生成し、それをキャロル402が利用できるようにした後、アリス401aは、少なくとも外部データ項目をキャロル402に送信する。

30

【0129】

次に、検証当事者であるキャロル402が行うアクションについて説明する。キャロル402は、アリス401aに、アリス401aが第1の署名のシェアを生成したことを証明してもらいたい。キャロル402は、第1の署名を取得する。アリス401aが第1の署名をキャロル402に送信し得るか、または第1の署名は、公的に入手可能であり、例えば、ブロックチェーン150上に記録され得る。第1の署名がブロックチェーントランザクションの入力に含まれる場合、キャロル402は、トランザクションから第1の署名を抽出することによって第1の署名を取得する。キャロル402はまた、アリス401aから候補の外部データ項目を取得する。ここで、「候補(の)(candidate)」は、アリス401aが、第1の署名内に、すなわちそれぞれの署名シェアに埋め込まれていると主張する外部データ項目を指すために使用される。実際にそうである場合、候補の外部データ項目は、上述した外部データ項目と同じである。しかしながら、この時点で、キャロル402は、そうであることを確認することができないので、「候補の」としている。

40

【0130】

キャロル402は、候補の外部データ項目を使用して、候補のエフェメラル公開鍵を生成する。これを行うために、キャロルは、アリスの外部データ項目(または、第1のデータ項目が外部データ項目だけでなくそれ以外にも基づいて生成される場合は第1のデータ項目)を取得し、対応する公開鍵を生成する。キャロル402はまた、他の参加者のデー

50

タ項目に対応する公開鍵を取得する。キャロル 402 は、候補を使用して候補のエフェメラル公開鍵を生成し、公開鍵を取得する。候補のエフェメラル公開鍵は、第 1 の構成要素および第 2 の構成要素、例えば、x 値および y 値を含む。

【0131】

キャロル 401 によって取得された第 1 の署名は、第 1 の署名構成要素および第 2 の署名構成要素を含む。アリス 401 a が第 1 の署名に寄与したことを検証するために、キャロル 402 は、候補の第 1 の署名構成要素を候補のエフェメラル公開鍵の第 1 の構成要素と比較する。それらが一致する場合、キャロル 402 は、アリス 401 a が実際に第 1 の署名のシェアを生成したことを確信することができる。すなわち、候補のエフェメラル公開鍵の第 1 の構成要素（候補の第 1 の署名構成要素と同等の x 構成要素を含む）と第 1 の署名構成要素とが一致するためには、候補の外部データ項目は、アリスの署名シェアを生成するために使用された外部データ項目でなければならない。アリス 401 a がキャロル 402 に候補の外部データ項目を提供したので、これは、アリス 401 a が署名のシェアを生成したことを証明する。このプロセスは、図 6 のステップ S 601 ~ S 605 に示されている。

10

【0132】

キャロル 402 はまた、対応する公開鍵に対して妥当性確認されたとき、第 1 の署名が有効な署名であることを検証し得る。第 1 の署名がブロックチェーントランザクションに署名するために使用され、そのトランザクションがブロックチェーンに記録されている場合、キャロル 401 は、第 1 の署名が有効な署名であると仮定し得る（すなわち、署名が有効でなかった場合、トランザクションはブロックチェーンノードによって受け入れられていなかったであろう）。しかしながら、キャロル 401 は、使用されているロック解除スクリプトが署名チェックを含むことを依然として検証し得る（すなわち、ブロックチェーンノードがトランザクション妥当性確認中に署名に対して署名チェックを実行したことを確認するために）。これを行うために、キャロル 401 は、使用されたトランザクションのロック解除スクリプトが OP_CHECKSIG スクリプトを含むことをチェックし得る。

20

【0133】

アリスの観点から本発明の実施形態を説明する際に上述したように、外部データ項目は、それ自体が、署名、すなわち第 2 の署名であってもよい。この場合、キャロル 402 は、例えばアリス 401 a から、第 2 のメッセージを取得し、例えば、証明された公開鍵などの、アリス 401 a によって提供されるかまたは他の方法でアリス 401 a にリンクされた公開鍵を使用して検証されたときに、第 2 の署名が有効な署名であることを検証し得る。

30

【0134】

アリス 401 a が、第 1 の署名を生成するために使用されるエフェメラル秘密鍵シェアを生成するためにソルト値を使用した場合、アリス 401 a は、ソルト値に対応する公開鍵をキャロル 402 に提供し得る。次いで、キャロル 402 は、「ソルト公開鍵」に基づいて、例えば、候補のエフェメラル公開鍵シェアとソルト公開鍵との組合せに基づいて、候補の第 1 の署名構成要素を生成し得る。上記組合せの x 値は、候補の第 1 の署名構成要素を生成するために使用され得る。一例を以下にさらに提供する。これらの例では、アリス 401 a はまた、キャロル 402 に第 3 の署名および第 3 のメッセージを提供し得る。キャロル 402 は、ソルト公開鍵を使用して検証されたときに、第 3 の署名が有効な署名であることを検証し得る。別のオプションの特徴として、アリス 401 a は、アリス 401 a がソルト値を知っていることを検証するためにキャロル 402 が使用し得る ZKP をキャロル 402 に提供し得る。

40

【0135】

以下に、ディーラーを伴わない ECDSA しきい値署名で使用するシークレットシェアにアイデンティティデータを埋め込む具体的な例を説明する。

【0136】

50

N人の参加者 $A_1, \dots, A_N$ のグループは、しきい値署名グループを形成することに同意する。修正された J V R S S (J V R S S - A) を実行するために必要なデータを生成するために、 $i \in \{1, \dots, N\}$ に対して、各参加者 401 が秘密鍵 $s_{k_{A_i}}$ (「主秘密鍵」) と、対応する公開鍵 $P_{K_{A_i}} = s_{k_{A_i}} \cdot G$ とを有すると仮定する。公開鍵 $P_{K_{A_i}}$ は、証明された公開鍵であってもよいし、証明された公開鍵から導出されてもよい。

【0137】

グループは、公開されており、後の時点で証明 (attestation) に使用される証明メッセージ M_{attest} (「第2のメッセージ」) に合意する。各グループメンバ 401 は、証明メッセージに署名して、「第2の署名」(すなわち、外部データ項目) を生成する：

【数39】

$$[r_{A_i}, s_{A_i}] = \text{Sig}(M_{attest})_{PK_{A_i}}, i \in \{1, \dots, N\}$$

10

【0138】

各参加者 401 はまた、自身の署名の (ダブル) ハッシュ

【数40】

$$\text{SHA256d}([r_{A_i}, s_{A_i}])$$

を計算し、これを、生成点を用いて難読化する：

20

【数41】

$$\text{SHA256d}([r_{A_i}, s_{A_i}]) \cdot G$$

【0139】

署名およびハッシュされたデータは、しきい値秘密分散および証明アルゴリズムの両方で使用される。

【0140】

N人の参加者間の共有シークレットは、秘密分散方式、例えば、上述した J V R S S 法を使用して生成することができる。J V R S S - A 変形の場合、アルゴリズムは、ゼロ次 (y 切片) を規定することによって修正される。以下の修正を加えて、J V R S S のステップ 1) ~ 5) に従う。

30

【0141】

ステップ 1) において、各参加者 401 は、(値をランダムに選択するのではなく) 自身のプライベート多項式のゼロ次を計算する：

【数42】

$$\begin{aligned} f_{A_1}(0) &= a_{10} = \text{SHA256d}([r_{A_1}, s_{A_1}]), \\ f_{A_2}(0) &= a_{20} = \text{SHA256d}([r_{A_2}, s_{A_2}]), \\ f_{A_3}(0) &= a_{30} = \text{SHA256d}([r_{A_3}, s_{A_3}]), \\ &\vdots \\ f_{A_N}(0) &= a_{N0} = \text{SHA256d}([r_{A_N}, s_{A_N}]), \end{aligned}$$

40

【0142】

他のすべての多項式係数はランダムに選択される。

【0143】

N人の参加者間の共有シークレットの逆数は、上述した I N V S S 方法を使用して計算することができる。I N V S S を使用することで、N個のグループエフェメラル鍵と、 $2t + 1$ 個の逆数シェアを生成することができる。証明セットアップを使用することによって、I N V S S は、I D 埋込みを組み込むように修正される。これは単に、共有エフェメ

50

ラル秘密鍵 k （むしろ、完全な秘密鍵自体は実際には生成されないので、共有エフェメラル秘密鍵のシェア）を生成するために、参加者 401 が、通常の J V R S S の代わりに J V R S S - A 法を使用するようにするためのものである。

【0144】

参加者はまた、多項式のゼロ次に対応する公開鍵を使用してマークルツリーを作成する。参加者は、以下のステップを行う。

1. 各参加者 401 は、J V R S S - A 法 $r_{i0} = k_{i0} \cdot G$ からの参加者の難読化された係数を使用して、以下のマークルツリーおよび対応するマークルルート R を計算することができ、ここで、各参加者 401 は、それらの署名の（ダブル）ハッシュに基づいて生成されたそれぞれのデータ項目 k_{i0} と、データ項目に対応するそれぞれの公開鍵 r_{i0} とを有する。

2. 次いで、参加者 401 は、この値をすべての参加者 401 にブロードキャストすることによって、全員が同じマークルルートを有することを確認することができる。その後、このマークルルートは、対応するエフェメラル鍵を使用して署名されるメッセージに含まれ得る。

【0145】

セットアップ：N 人の独立した参加者 $A_1, \dots, A_N$ は、しきい値秘密分散グループへの参加に同意する。その参加者らはまた、将来の証明を可能にするトラストレスセットアップを使用することに同意する。参加者は、N 人の参加者のグループのための証明メッセージセットアップを使用するとともに（ r_{i0} の和に基づいて）証明メッセージ依存グループエフェメラル鍵 r を生成する。証明メッセージが、署名されているトランザクションになることをグループが望む場合、このステップは、一旦メッセージが知られると、署名の作成中に完了される必要があることに留意されたい。

【0146】

各参加者 401 は、J V R S S に従って、秘密鍵シェア $a_i = J V R S S(i)$ を計算する。公開鍵は、プライベート多項式の難読化されたゼロ次値を加算することによって計算される：

【数 43】

$$PK = (f_{A_1}^a(0) \cdot G) + (f_{A_2}^a(0) \cdot G) + \dots + (f_{A_N}^a(0) \cdot G)$$

【0147】

しきい値公開鍵および秘密鍵シェア導出は修正されないことに留意されたい。秘密鍵シェア導出は、証明なしで J V R S S に従うべきである。

【0148】

$2t+1$ 人の参加者 $A_1, \dots, A_{2t+1}$ のサブセットが、 PK_B に支払うトランザクションに署名したいと望む。

【0149】

その参加者らは、署名入力の一部として、証明メッセージ依存エフェメラル鍵 r を使用して、上述した「非最適署名生成」のステップ 1) ~ 7) に従う。代わりに最適な署名生成方法が使用されてもよいことに留意されたい。以下に概要を示す。

【0150】

すべての署名参加者 401 がトランザクション $T x_{threshold}$ （以下の表を参照）について合意する。

10

20

30

40

【表 1】

$Tx'_{threshold}$			
入力		出力	
値	ScriptSig	値	ScriptPubkey
1 BSV		0.99 BSV	OP_DUP OP_HASH160 < $H_{160}(PK_B)$ > OP_EQUALVERIFY OP_CHECKSIG
		0 BSV	OP_FALSE OP_RETURN R

10

【0151】

各参加者401は、メッセージダイジェスト $e = \text{SHA-256d}(Tx'_{threshold})$ を計算する。

【0152】

各参加者401は、署名シェア

【数44】

$$s_i = k_i^{-1}(e + a_i r) \bmod n$$

20

を計算し、ここで、

【数45】

$$k_i^{-1}$$

(それぞれのエフェメラル秘密鍵シェアの逆数) および r (共有エフェメラル秘密鍵に対応する公開鍵) は、上述した方法を使用して導出される。

30

【0153】

コーディネータ404は、署名シェアを収集し、補間する：

【数46】

$$s = \text{interpolate}(s_1, \dots, s_{2t+1})$$

【0154】

出力(「第1の署名」)は、 $[r, s]$ であり、トランザクションは署名することができる(完全なトランザクションについては以下の表を参照)。

【表 2】

40

$Tx_{threshold}$			
入力		出力	
値	ScriptSig	値	ScriptPubkey
1 BSV	< $[r, s]$ > < PK >	0.99 BSV	OP_DUP OP_HASH160 < $H_{160}(PK_B)$ > OP_EQUALVERIFY OP_CHECKSIG

50

【 0 1 5 5 】

参加者 $A_1, \dots, A_N$ のアイデンティティは、この時点で、エフェメラル鍵 r に埋め込まれており、したがって、第 1 の署名に埋め込まれている。 M_{attest} （これは、提供されてもよいし、公に知られていてもよい）が与えられると、参加者 4 0 1 は、自身がしきい値グループのメンバであることを証明することができる。

【 0 1 5 6 】

アリス 4 0 1 a が、秘密アイデンティティ鍵 $s k_{A1}$ を有する参加者 A_1 であると仮定する。値 M_{attest} は公開されており、アリス 4 0 1 a は、第 2 の署名を生成することができる：

【数 4 7】

$$[r_{A_1}, s_{A_1}] = \text{Sig}(M_{attest})_{PK_{A1}}$$

10

【 0 1 5 7 】

グループ公開エフェメラル鍵 r を生成するためのアルゴリズムを実行した結果、アリスはまた、各データ項目に対応する公開鍵

【数 4 8】

$$\{k_{20} \cdot G, \dots, k_{N,0} \cdot G\}$$

20

と、 $T x_{threshold}$ に埋め込まれる対応するマークルルート R とを保持する。

【 0 1 5 8 】

検証者 4 0 2 は、第 1 の署名 $[r, s]$ を有する $T x_{threshold}$ および参加者の数 N を知る。

【 0 1 5 9 】

証明プルーフを実行するために、アリス 4 0 1 a は、最初に、検証者であるキャロル 4 0 2 に下記を提供する：

【数 4 9】

$$[r_{A_1}, s_{A_1}] = \text{Sig}(M_{attest})_{PK_{A1}}, \{k_{20} \cdot G, \dots, k_{N0} \cdot G\}$$

30

【 0 1 6 0 】

キャロル（検証者）は、候補のエフェメラル公開鍵（ x 構成要素および y 構成要素を含む）を計算する：

【数 5 0】

$$(x', y') = \text{SHA256d}([r_{A_1}, s_{A_1}]) \cdot G + k_{20} \cdot G + \dots + k_{N0} \cdot G$$

また、候補のエフェメラル公開鍵の x 構成要素に基づいて候補の第 1 の署名構成要素を計算する：

40

【数 5 1】

$$r' = x' \bmod n$$

そして、 $r' = r$ であること（すなわち、候補の第 1 の署名構成要素が第 1 の署名構成要素と一致すること）をチェックする。

【 0 1 6 1 】

キャロルはまた、 $[r, s]$ が、グループ公開鍵 PK （ここで、 $PK = a \cdot G$ である）を使用して妥当性確認されたときに、有効な署名であることを検証する。

【 0 1 6 2 】

50

以下では、マークルプルーフ証明について説明する。キャロル 402 はマークルルート R を知っている と 仮定する。アリスは、マークルプルーフを生成して、
【数 5 2】

$$[SHA256d([r_{A_1}, s_{A_1}]) \cdot G]$$

が、R によって表される集合の要素であることを立証する。アリス 401 a は、セットアップ中に他のグループメンバからすべてのエフェメラル鍵シェアを受信し終わっているので、マークルツリー全体を再生成してプルーフを生成することができることに留意されたい。

10

【0163】

検証者がすべての参加者を訪ね、共有エフェメラル秘密鍵 k を作成するのに十分な情報を入手し、次いで、その結果を署名とともに使用して共有秘密鍵を導出するという攻撃がある。これを防止するために、アリス 401 a は、秘密のソルト値を含めることができる。明示的に、ゼロ次の参加者 i のエフェメラル鍵は以下のように設定され得る：

【数 5 3】

$$f_{A_i}(0) = k_{i0} = SHA256d([r_{A_1}, s_{A_1}]) + w$$

【0164】

次いで、アリス 401 a は、ゼロ知識証明を提供することによって、または w で生成された署名を提供することによって、これの知識を証明することができる。このソルトは秘密にしておくべきである。

20

【0165】

この方法は、メッセージがトランザクションであることに関連して説明されているが、署名が行われるメッセージは、これに限定される必要はないことに留意されたい。

【0166】

以下は、説明される方法のいくつかのセキュリティ面の考察である。

【0167】

エフェメラル鍵シェア k_{A_1} をマスクするためのソルト値の使用は、スプーフィング攻撃に対する保護として機能する。仮にエフェメラル鍵が第 1 のトランザクション署名のみに依存する場合（追加のランダム性なし）、敵対者は、アリス（証明者）から受信した情報を単に保持することによって他者にプルーフをリプレイすることが可能であり得、元のトランザクション署名者を、プルーフを見ただけの人と区別することを困難にする。

30

【0168】

署名対署名マップにおいて 1 回限りのシークレット値の使用を採用することで、すべての秘密鍵が安全であるだけでなく、敵対者がアリス 401 a になりすますために取得したいかなる情報も使用することができないことが保証される。したがって、この方法は、キャロル（検証者）402 が正直な当事者であることも信頼できる当事者であることも必要としない。

40

【0169】

しきい値署名者プルーフは、マークルプルーフが提供されない限り、スプーフィングに対して脆弱であり得る。マークルプルーフが証明の一部として含まれていない場合、および証明メッセージが公に知られている場合、非グループメンバは、プルーフデータをリバースエンジニアリングすることができる。

【0170】

イブ（盗聴者）が、 (x', y') （ここで、 $r' = x' \bmod n$ ）と、グループ証明メッセージ M_{attest} とを知っている場合、イブは、自分自身の秘密鍵 s_{k_E} を取って、署名

【数 5 4】

50

$$[r_E, s_E] = \text{Sig}(M_{\text{attest}})_{PK_E}$$

を生成することができる。

【0171】

次いで、イブは、 $U = (x', y') - \text{SHA256d}([r_E, s_E]) \cdot G$ を計算することができ、 U を部分 $U_1, U_2, \dots, U_{N-1}$ （ここで、 $U = U_1 + \dots + U_N$ である）に自明に分割する。

【0172】

イブは、計算されたデータを使用して、自身がグループ署名イベントの参加者であったと装うことができる。実際のグループ署名者から秘密情報が漏洩することはないが、第三者が、グループ署名イベントに参加したと装うことはできる。

【0173】

グループによって署名されたメッセージにマークルルートを含めることによって、このスプーフィングが不可能になる。

【0174】

結論

開示された技法の他の変形または使用事例は、本明細書の開示が与えられると、当業者には明らかになり得る。本開示の範囲は、記載された実施形態によって限定されず、添付の特許請求の範囲によってのみ限定される。

【0175】

例えば、上記のいくつかの実施形態は、ビットコインネットワーク106、ビットコインブロックチェーン150、およびビットコインノード104に関して説明されている。しかしながら、ビットコインブロックチェーンはブロックチェーン150の1つの特定の例であり、上記の説明は一般に任意のブロックチェーンに適用され得ることが理解されよう。すなわち、本発明は、決してビットコインブロックチェーンに限定されるものではない。より一般的には、ビットコインネットワーク106、ビットコインブロックチェーン150、およびビットコインノード104への上記のいかなる言及も、それぞれ、ブロックチェーンネットワーク106、ブロックチェーン150、およびブロックチェーンノード104への言及に置き換えられ得る。ブロックチェーン、ブロックチェーンネットワーク、および／またはブロックチェーンノードは、上述したビットコインブロックチェーン150、ビットコインネットワーク106、およびビットコインノード104の説明された特性のいくつかまたはすべてを共有し得る。

【0176】

本発明の好ましい実施形態では、ブロックチェーンネットワーク106は、ビットコインネットワークであり、ビットコインノード104は、ブロックチェーン150のブロック151を作成し、公開し、伝搬し、記憶するという説明した機能のすべてを少なくとも実行する。これらの機能のうちの1つまたはいくつかのみを実行し、すべては実行しない他のネットワークエンティティ（またはネットワーク要素）が存在し得ることは除外されない。すなわち、ネットワークエンティティは、ブロックを作成および公開することなく、ブロックを伝搬および／または記憶する機能を実行し得る（これらのエンティティは、好ましいビットコインネットワーク106のノードとはみなされないことを想起されたい）。

【0177】

本発明の好ましくない実施形態では、ブロックチェーンネットワーク106は、ビットコインネットワークでない可能性がある。これらの実施形態では、ノードが、ブロックチェーン150のブロック151を作成し、公開し、伝搬し、記憶するという機能のうちの少なくとも1つまたはいくつかを実行し、すべては実行しない可能性があることは除外されない。例えば、それらの他のブロックチェーンネットワーク上で、「ノード」は、ブロック151を作成して公開するが、それらのブロック151を記憶および／または他のノ

10

20

30

40

50

ードに伝搬しないように構成されるネットワークエンティティを指すために使用され得る。

【0178】

さらにより一般的には、上記の「ビットコインノード」104という用語へのいかなる言及も、「ネットワークエンティティ」または「ネットワーク要素」という用語に置き換えられ得、そのようなエンティティ／要素は、ブロックの作成、公開、伝搬、および記憶という役割のうちのいくつかまたはすべてを実行するように構成される。そのようなネットワークエンティティ／要素の機能は、ブロックチェーンノード104を参照して上述したのと同じ方法でハードウェアに実装され得る。

【0179】

上記の実施形態は、単に例として説明されていることが理解されよう。より一般的には、下記ステートメントのうちのいずれか1つまたは複数による方法、装置、またはプログラムが提供され得る。

【0180】

ステートメント1. デジタル署名のシェアを生成するコンピュータ実装方法であって、参加者のグループの各参加者は、第1の共有秘密鍵のそれぞれのシェアを有し、方法は、グループの第1の参加者によって実行され、

第1のメッセージを取得することと、

少なくとも第1の外部データ項目のハッシュに基づいて第1のデータ項目を生成することと、

エフェメラル秘密鍵の第1のエフェメラル秘密鍵シェアを生成することであって、第1のエフェメラル秘密鍵シェアは、第1のデータ項目と、各他の参加者によって生成されたそれぞれのデータ項目とに基づいて生成される、ことと、

エフェメラル秘密鍵に対応するエフェメラル公開鍵を生成することと、

第1のメッセージと、第1のエフェメラル秘密鍵シェアと、第1の共有秘密鍵の第1のシェアと、エフェメラル公開鍵とに基づいて、第1の署名シェアを生成することと、

少なくとももしきい値数のそれぞれの署名シェアに基づいて第1の署名を生成するために、第1の署名シェアをコーディネータが利用できるようにすることと

を含む方法。

【0181】

外部データ項目は、第1の当事者の識別子、例えば、名前、住所、電話番号、国民保険番号、パスポート番号、公開鍵などを含み得る。

【0182】

好ましくは、第1の署名は、ECDSA署名である。「エフェメラル公開鍵」は、「エフェメラル公開鍵のx構成要素」の省略表現として使用される。

【0183】

好ましくは、各他の参加者は、同じ方法で、それぞれの外部データ項目に基づいて、それぞれのエフェメラル公開鍵シェアを生成する。

【0184】

ステートメント2. 第1のエフェメラル秘密鍵シェアを生成することは、他の参加者の各々との秘密分散方式を実行することを含む、ステートメント1に記載の方法。

【0185】

ステートメント3. 秘密分散方式は、共同検証可能秘密分散(JVRS S)方式である、ステートメント2に記載の方法。

【0186】

より正確には、秘密分散方式は、各参加者がランダムな多項式係数を生成する代わりに、各参加者が第1の外部データ項目のハッシュとしてゼロ次係数を設定するJVRS S方式の修正バージョンであり得る。ランダムな多項式係数は、真にランダムまたは擬似ランダムであり得る(例えば、任意に選択され得る)。

【0187】

10

20

30

40

50

ステートメント 4. 第 1 のデータ項目は、第 1 の多項式のゼロ次係数であり、J V R S S 方式は、

第 1 の多項式のそれぞれのインスタンスを他の参加者の各々に送信することであって、第 1 の多項式のそれぞれのインスタンスは、それぞれの参加者のそれぞれのインデックスに基づいて生成される、ことと、

各他の参加者からそれぞれの多項式を取得することであって、それぞれの多項式は、第 1 の参加者のそれぞれのインデックスと、他の参加者によって生成されたそれぞれのデータ項目とに基づいて生成される、ことと

を含む、ステートメント 3 に記載の方法。

【0188】

ステートメント 5. エフェメラル秘密鍵に対応するエフェメラル公開鍵を生成することは、

第 1 のデータ項目に対応する第 1 の公開鍵を生成することと、

各他の参加者から、その他の参加者によって生成されたそれぞれのデータ項目に対応するそれぞれの公開鍵を取得することと、

を含む、先行ステートメントのいずれかに記載の方法。

【0189】

ステートメント 6. 第 1 の参加者が第 1 の署名を生成したことを証明するために、第 1 の外部データ項目と、それぞれのデータ項目に対応する取得されたそれぞれの公開鍵とを検証当事者が利用できるようにすることを含む、ステートメント 5 に記載の方法。

【0190】

ステートメント 7. 第 1 のメッセージを取得することは、第 1 のメッセージを生成することを含み、方法は、第 1 のメッセージを検証当事者が利用できるようにすることを含む、先行ステートメントのいずれかに記載の方法。

【0191】

ステートメント 8.

第 2 のメッセージを取得することと、

少なくとも第 2 のメッセージと第 1 の参加者の主秘密鍵とに基づいて第 2 の署名を生成することと

を含み、外部データ項目は第 2 の署名を含む、

先行ステートメントのいずれかに記載の方法。

【0192】

ステートメント 9. 各参加者は、同じ第 2 のメッセージに基づいて、それぞれの外部データ項目を生成する、ステートメント 8 に記載の方法。

【0193】

ステートメント 10. 第 1 の参加者の主秘密鍵は、第 1 の参加者のアイデンティティにリンクされた主公開鍵に対応する、ステートメント 8 またはステートメント 9 に記載の方法。

【0194】

ステートメント 11. 第 1 の共有秘密鍵の第 1 のシェアは、秘密分散方式を使用して生成される、先行ステートメントのいずれかに記載の方法。

【0195】

例えば、J V R S S である。

【0196】

ステートメント 12. 第 1 の参加者はコーディネータであり、方法は、

少なくともしきい値数のそれぞれの署名シェアを受信することと、

第 1 の署名構成要素および第 2 の署名構成要素を含む第 1 の署名を生成することと

を含み、第 1 の署名構成要素は、エフェメラル公開鍵に基づいて生成され、第 2 の署名構成要素は、少なくともしきい値数のそれぞれの署名シェアに基づいて生成される、先行ステートメントのいずれかに記載の方法。

10

20

30

40

50

【0197】

ステートメント13. 第1のデータ項目は、ランダムソルト値に基づいて生成される、先行ステートメントのいずれかに記載の方法。

【0198】

ステートメント14. 方法は、ランダムソルト値の知識を証明するためのゼロ知識証明を検証当事者に提供することを含む、ステートメント13に記載の方法。

【0199】

ステートメント15. ランダムソルト値は秘密鍵であり、方法は、第3のメッセージを取得することと、

少なくともランダムソルト値と第3のメッセージとに基づいて第3の署名を生成することと、 10

ランダムソルト値に対応する公開鍵を使用して検証されたときに、第3の署名が第3のメッセージの有効な署名であることを証明するために、第3の署名と、第3のメッセージと、ランダムソルト値に対応する公開鍵とを検証当事者が利用できるようにすることを含む、ステートメント13またはステートメント14に記載の方法。

【0200】

ステートメント16. 第3のメッセージは第2のメッセージを含む、ステートメント15に記載の方法。

【0201】

例えば、第3のメッセージは、第2のメッセージと同じであってもよい。 20

【0202】

ステートメント17.

ハッシュツリーのルートハッシュを生成することであって、それぞれのデータ項目に対応する各それぞれの公開鍵が、ハッシュツリーのそれぞれのリーフハッシュを生成するためにハッシュされる、ことと、

ハッシュツリーのルートに参加者のうちの1つまたは複数および／または検証当事者に送信することと

を含む、先行ステートメントのいずれかに記載の方法。

【0203】

ステートメント18. 第1のデータ項目に対応する第1の公開鍵がハッシュツリーの要素であることを検証するために、ハッシュプルーフを検証当事者に送信することを含む、ステートメント15に記載の方法。 30

【0204】

好ましくは、ハッシュツリーはマークルツリーであり、ハッシュルートはマークルルートであり、ハッシュプルーフはマークルプルーフである。

【0205】

ステートメント19. 第1のメッセージは、ブロックチェーントランザクションの少なくとも一部を含む、先行ステートメントのいずれかに記載の方法。

【0206】

例えば、第1の署名は、ブロックチェーントランザクションに署名するために使用され得る。 40

【0207】

ステートメント20. ブロックチェーントランザクションはハッシュルートを含む、ステートメント18およびステートメント19の方法。

【0208】

ステートメント21. ブロックチェーントランザクションは、第1の署名を含む、ステートメント19またはステートメント20に記載の方法。

【0209】

ステートメント22. 第1のメッセージを検証当事者が利用できるようにすることは、ブロックチェーントランザクションをブロックチェーンネットワークに送信することを含 50

む、ステートメント 19 から 21 のいずれかに記載の方法。

【0210】

ステートメント 23．外部データ項目のハッシュは、外部データ項目のダブルハッシュである、先行ステートメントのいずれかに記載の方法。

【0211】

ステートメント 24．デジタル署名が第 1 の参加者によって部分的に生成されたことを検証するコンピュータ実装方法であって、方法は、検証当事者によって実行され、

第 1 の署名構成要素および第 2 の署名構成要素を含む第 1 の署名を取得することと、

第 1 の参加者からの候補の第 1 の外部データ項目と、それぞれのデータ項目に対応する 1 つまたは複数のそれぞれの公開鍵とを、他の参加者ごとに 1 つずつ取得することと、

候補の第 1 の外部データ項目のハッシュに基づいて候補の公開鍵を生成することと、
候補の公開鍵と、取得された 1 つまたは複数の公開鍵とに基づいて、候補のエフェメラル公開鍵を生成することと、

候補のエフェメラル公開鍵に基づいて候補の第 1 の署名構成要素を生成することと、

候補の第 1 の署名構成要素が第 1 の署名構成要素に対応するかどうかに基づいて、第 1 の署名が第 1 の参加者によって部分的に生成されたことを検証することと

を含む方法。

【0212】

ステートメント 25．第 1 の署名は第 1 のメッセージに署名し、方法は、

第 1 の署名を生成するために使用された共有秘密鍵に対応する共有公開鍵を取得することと、

第 1 の公開鍵を使用して検証されたときに、第 1 の署名が第 1 のメッセージの有効な署名であることを検証することと

を含む、ステートメント 24 に記載の方法。

【0213】

ステートメント 26．候補の外部データ項目は第 2 の署名である、ステートメント 24 またはステートメント 25 に記載の方法。

【0214】

ステートメント 27．第 2 の署名は第 2 のメッセージに署名し、方法は、

第 2 の署名を生成するために使用された秘密鍵に対応する第 2 の公開鍵を取得することと、

第 2 の公開鍵を使用して検証されたときに、第 2 の署名が第 2 のメッセージの有効な署名であることを検証することと

を含む、ステートメント 24 から 26 のいずれかに記載の方法。

【0215】

ステートメント 28．候補の公開鍵は、ランダムソルト値に基づいて生成され、方法は、

ランダムソルト値の知識を証明するためのゼロ知識証明を受信することと、

第 1 の参加者がランダムソルト値の知識を有することを検証することと

を含む、ステートメント 24 またはステートメント 27 に記載の方法。

【0216】

ステートメント 29．候補の公開鍵は、ランダムソルト値に基づいて生成され、方法は、

第 3 のメッセージを取得することと、

第 3 の署名を取得することと、

ランダムソルト値に対応する公開鍵を使用して検証されたときに、第 3 の署名が第 3 のメッセージの有効な署名であることを検証することと

を含む、ステートメント 24 から 28 のいずれかに記載の方法。

【0217】

ステートメント 30．

10

20

30

40

50

ハッシュブーフおよびハッシュルートを取得することであって、第1のメッセージはハッシュルートを含む、ことと、

ハッシュブーフに基づいて、候補の第1のエフェメラル公開鍵シェアが、ハッシュルートを含むハッシュツリーの要素であることを検証することとを含む、ステートメント24から29のいずれかに記載の方法。

【0218】

ステートメント31. 第1の署名は、第1の当事者から受信される、ステートメント24から30のいずれかに記載の方法。

【0219】

ステートメント32. 第1のメッセージは、第1の参加者から受信される、ステートメント24から31のいずれかに記載の方法。

【0220】

ステートメント33. 第1のメッセージは、検証当事者によって生成される、ステートメント24から31のいずれかに記載の方法。

【0221】

ステートメント34. 第1のメッセージは、ブロックチェーントランザクションの少なくとも一部を含む、ステートメント24から33のいずれかに記載の方法。

【0222】

ステートメント35. 第1のメッセージを取得することは、ブロックチェーンからブロックチェーントランザクションを取得することを含む、ステートメント34に記載の方法。

【0223】

ステートメント36. 第1の署名を取得することは、ブロックチェーントランザクションから第1の署名を抽出することを含む、ステートメント35に記載の方法。

【0224】

ステートメント37. ブロックチェーントランザクションの入力は、第1の署名を含み、方法は、

ブロックチェーントランザクションの入力によって参照される前のブロックチェーントランザクションの出力が署名検証スクリプトを含むことを検証すること

を含む、ステートメント33から36のいずれかに記載の方法。

【0225】

ステートメント38. コンピュータ機器であって、

1つまたは複数のメモリユニットを備えるメモリと、

1つまたは複数の処理ユニットを備える処理装置と、

を備え、メモリは、処理装置上で実行されるように構成されたコードを記憶し、コードは、処理装置上にあるときに、ステートメント1から37のいずれかに記載の方法を実行するように構成される、

コンピュータ機器。

【0226】

ステートメント39. コンピュータ可読ストレージ上に具現化されたコンピュータプログラムであって、1つまたは複数のプロセッサ上で実行されると、ステートメント1から37のいずれかに記載の方法を実行するように構成されたコンピュータプログラム。

【0227】

本明細書で開示される別の態様によれば、第1の参加者および検証当事者のアクションを含む方法が提供され得る。

【0228】

本明細書で開示される別の態様によれば、第1の参加者および検証当事者のコンピュータ機器を備えるシステムが提供され得る。

10

20

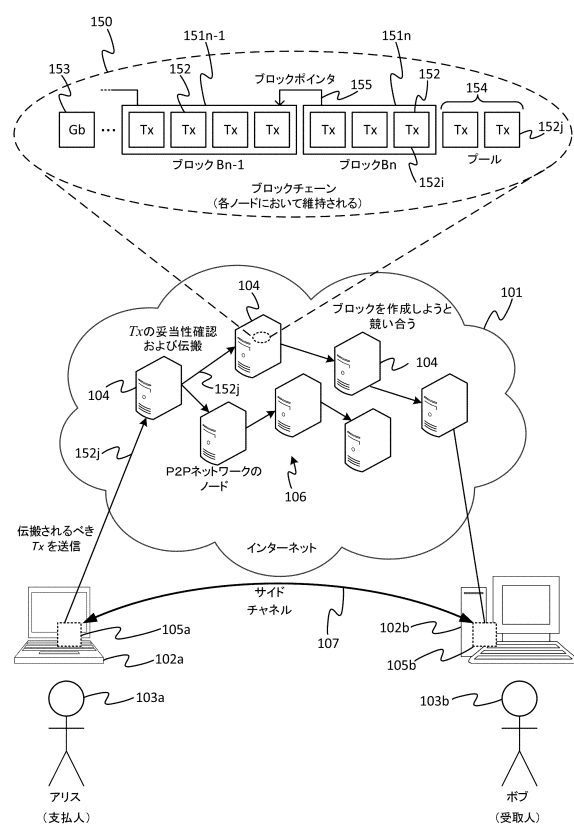
30

40

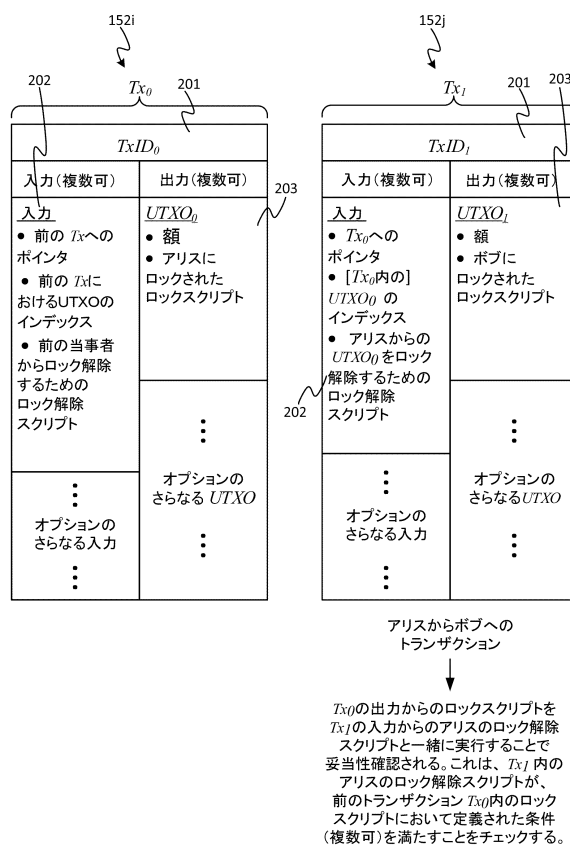
50

【図面】

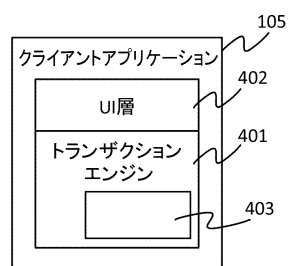
【図 1】



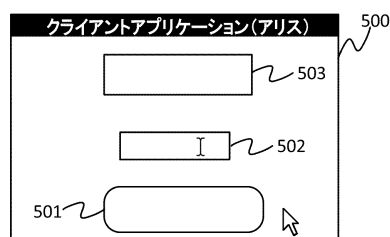
【図 2】



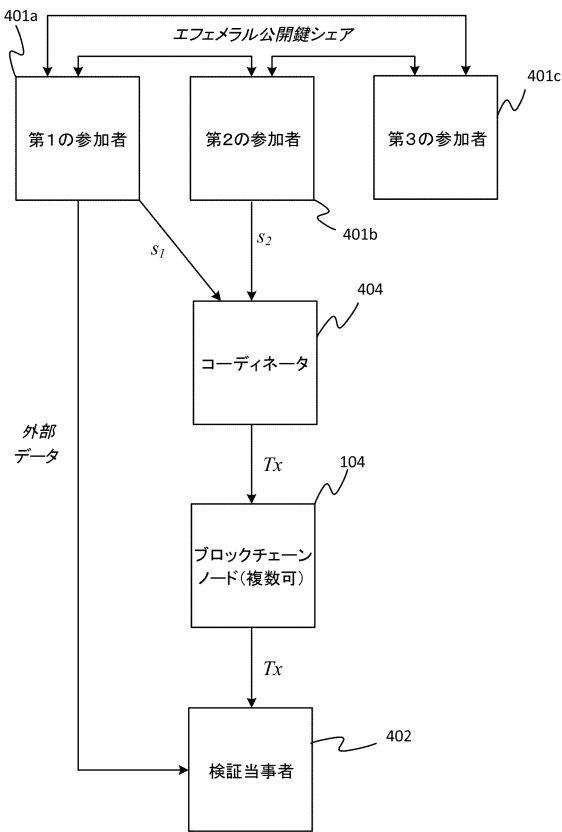
【図 3 A】



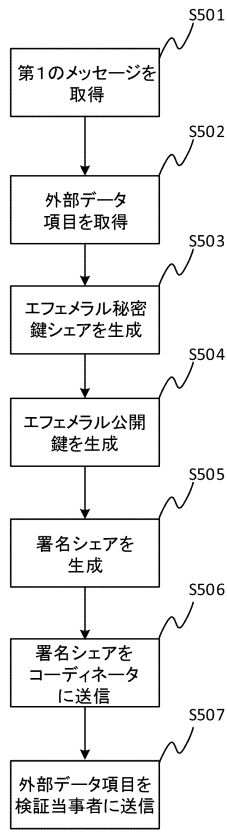
【図 3 B】



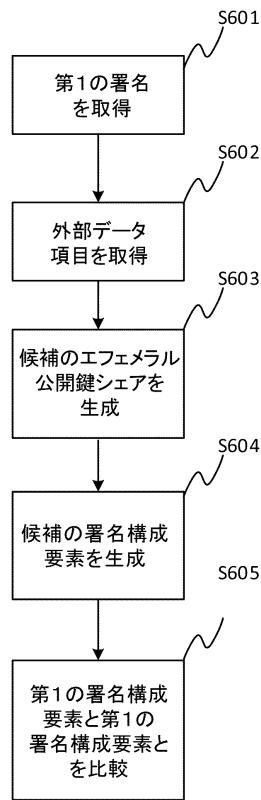
【図 4】



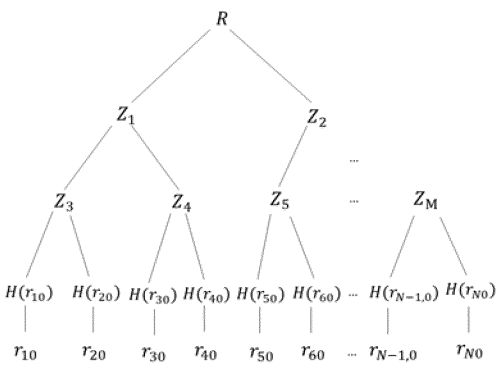
【図 5】



【図 6】



【図 7】



10

20

30

40

50

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2021/070124

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04L9/32 H04L9/08
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2019/034986 A1 (NCHAIN HOLDINGS LTD [AG]) 21 February 2019 (2019-02-21) page 2, line 1 - page 10, line 10 page 44, line 5 - page 51, line 25 -----	1-39
Y	WO 2019/193452 A1 (NCHAIN HOLDINGS LTD [AG]) 10 October 2019 (2019-10-10) abstract page 10, line 1 - page 15, line 6 page 20, line 14 - page 33, line 15 -----	1-39
Y	WO 2017/145016 A1 (NCHAIN HOLDINGS LTD [AG]) 31 August 2017 (2017-08-31) cited in the application abstract paragraph [0009] - paragraph [0042] paragraph [0084] - paragraph [0106] paragraph [0119] - paragraph [0120] paragraph [0140] - paragraph [0159] -----	1-39

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

8 October 2021

Date of mailing of the international search report

18/10/2021

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040
Fax: (+31-70) 340-3016

Authorized officer

Apostolescu, Radu

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2021/070124

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2019034986 A1	21-02-2019	NONE	
WO 2019193452 A1	10-10-2019	CN 112204920 A EP 3777013 A1 JP 2021519541 A KR 20200139223 A SG 11202008860Y A TW 201944755 A US 2021152371 A1 WO 2019193452 A1	08-01-2021 17-02-2021 10-08-2021 11-12-2020 29-10-2020 16-11-2019 20-05-2021 10-10-2019
WO 2017145016 A1	31-08-2017	AU 2017223133 A1 BR 112018016245 A2 CA 3010116 A1 CL 2018002363 A1 CN 108292402 A CO 2018009236 A2 DK 3268914 T3 EP 3268914 A1 EP 3364598 A1 EP 3751783 A1 ES 2687182 T3 GB 2561728 A HK 1251840 A1 HR P20181373 T1 HU E040631 T2 IL 260895 A JP 6515246 B2 JP 2019507510 A KR 20180116278 A LT 3268914 T PH 12018550109 A1 PL 3268914 T3 PT 3268914 T SG 10201805995V A SG 11201805472R A SI 3268914 T1 TW 201733303 A TW 202127831 A US 2019052458 A1 US 2020274700 A1 WO 2017145016 A1 ZA 201708483 B	09-08-2018 18-12-2018 31-08-2017 23-11-2018 17-07-2018 22-11-2018 08-10-2018 17-01-2018 22-08-2018 16-12-2020 24-10-2018 24-10-2018 12-04-2019 11-01-2019 28-03-2019 31-08-2021 15-05-2019 14-03-2019 24-10-2018 12-11-2018 18-03-2019 31-12-2018 11-10-2018 30-08-2018 30-07-2018 30-11-2018 16-09-2017 16-07-2021 14-02-2019 27-08-2020 31-08-2017 26-06-2019

10

20

30

40

50

フロントページの続き

MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,N
E,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,
CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,IT,JO,JP,K
E,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,N
G,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,ST,SV,SY,TH,TJ,TM,
TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,WS,ZA,ZM,ZW

(72)発明者 ライト，クレイグ，スティーヴン
イギリス ダブリュー1ダブリュー 8エーピー ロンドン マーケット プレイス 30 エヌチェー
ン ライセンシング アーゲー 内

(72)発明者 マッケイ，アレクサンダー
イギリス ダブリュー1ダブリュー 8エーピー ロンドン マーケット プレイス 30 エヌチェー
ン ライセンシング アーゲー 内